

Docket No. 220963US2S/btm



2171
RECEIVED

JUL 09 2002

Technology Center 2100
#4

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Mari OMORI, et al.

GAU: 2171

SERIAL NO: 10/098,543

EXAMINER:

FILED: March 18, 2002

FOR: METHOD, COMPUTER PROGRAM PRODUCT AND SYSTEM FOR PROVIDING INFORMATION
PROCESSING SERVICE

REQUEST FOR PRIORITY

ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231

SIR:

- ☐ Full benefit of the filing date of U.S. Application Serial Number [US App No], filed [US App Dt], is claimed pursuant to the provisions of **35 U.S.C. §120**.
- ☐ Full benefit of the filing date of U.S. Provisional Application Serial Number , filed , is claimed pursuant to the provisions of **35 U.S.C. §119(e)**.
- ☒ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of **35 U.S.C. §119**, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

<u>COUNTRY</u>	<u>APPLICATION NUMBER</u>	<u>MONTH/DAY/YEAR</u>
JAPAN	2001-079353	March 19, 2001
JAPAN	2002-069188	March 13, 2002

Certified copies of the corresponding Convention Application(s)

- ☒ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee
- ☐ were filed in prior application Serial No. filed
- ☐ were submitted to the International Bureau in PCT Application Number .
Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.
- ☐ (A) Application Serial No.(s) were filed in prior application Serial No. filed ; and
(B) Application Serial No.(s)
 - ☐ are submitted herewith
 - ☐ will be submitted prior to payment of the Final Fee

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.


Marvin J. Spivak

Registration No. 24,913

Surinder Sachar
Registration No. 34,423



22850

Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 10/98)

10/098,543

日 本 国 特 許 庁
JAPAN PATENT OFFICE



別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office

出 願 年 月 日

Date of Application:

2002年 3月13日

出 願 番 号

Application Number:

特願2002-069188

ST.10/C]:

[JP2002-069188]

出 願 人
Applicant(s):

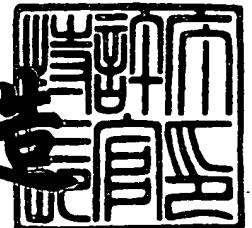
株式会社東芝

CERTIFIED COPY OF
PRIORITY DOCUMENT

2002年 4月 5日

特 許 庁 長 官
Commissioner,
Japan Patent Office

及 川 耕 造



【書類名】 特許願

【整理番号】 A000200970

【提出日】 平成14年 3月13日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/00

【発明の名称】 情報処理サービス提供方法及びプログラム並びにシステム

【請求項の数】 17

【発明者】

 【住所又は居所】 東京都府中市東芝町1番地 株式会社東芝府中事業所内

 【氏名】 海邊 麻理

【発明者】

 【住所又は居所】 東京都府中市東芝町1番地 株式会社東芝府中事業所内

 【氏名】 松本 茂

【発明者】

 【住所又は居所】 東京都府中市東芝町1番地 株式会社東芝府中事業所内

 【氏名】 守安 隆

【発明者】

 【住所又は居所】 東京都府中市東芝町1番地 株式会社東芝府中事業所内

 【氏名】 佐波 公夫

【特許出願人】

 【識別番号】 000003078

 【氏名又は名称】 株式会社 東芝

【代理人】

 【識別番号】 100058479

 【弁理士】

 【氏名又は名称】 鈴江 武彦

 【電話番号】 03-3502-3181

【選任した代理人】

【識別番号】 100084618

【弁理士】

【氏名又は名称】 村松 貞男

【選任した代理人】

【識別番号】 100068814

【弁理士】

【氏名又は名称】 坪井 淳

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【先の出願に基づく優先権主張】

【出願番号】 特願2001- 79353

【出願日】 平成13年 3月19日

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705037

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理サービス提供方法及びプログラム並びにシステム

【特許請求の範囲】

【請求項 1】 コンピュータによる情報処理サービス提供方法において、
ネットワークを経由して提供される情報処理サービスに対する付加的な情報処理サービスを依頼するイベントを検出し、

前記イベントが検出された場合に、前記付加的な情報処理サービスを提供する付加的機能の呼び出しを定義したライブラリを参照し、

前記ライブラリの参照結果に基づいて、前記付加的な情報処理サービスを提供する付加的機能に実行を依頼する

ことを特徴とする情報処理サービス提供方法。

【請求項 2】 コンピュータによる情報処理サービス提供方法において、
ネットワークを経由して提供される情報処理サービスの運用状態を示すデータを記録し、

前記情報処理サービスに対する付加的な情報処理サービスを提供する付加的機能に対して、記録された前記運用状態を示すデータを提供する

ことを特徴とする情報処理サービス提供方法。

【請求項 3】 コンピュータによる情報処理サービス提供方法において、
ネットワークを経由して提供される情報処理サービスのリクエストを取得し、
前記リクエストに応じて、前記情報処理サービスに対する付加的な情報処理サービスを提供する付加的機能に実行を依頼する

ことを特徴とする情報処理サービス提供方法。

【請求項 4】 コンピュータによる情報処理サービス提供方法において、
ネットワークを経由して提供される情報処理サービスの処理に基づいて発生する前記情報処理サービスに対する付加的な情報処理サービスのリクエストを取得し、

前記リクエストに応じて、前記付加的な情報処理サービスを提供する付加的機能に実行を依頼する

ことを特徴とする情報処理サービス提供方法。

【請求項 5】 請求項 3 又は請求項 4 記載の情報処理サービス提供方法において、

前記付加的な情報処理サービスを提供する付加的機能に実行を依頼するために、前記リクエストと前記付加的機能とを対応付けたテーブルを参照し、前記リクエストに対応する前記付加的機能を特定することを特徴とする情報処理サービス提供方法。

【請求項 6】 請求項 1 乃至請求項 5 のいずれか記載の情報処理サービス提供方法において、

前記付加的機能は、複数の情報処理サービスに共同で使用されることを特徴とする情報処理サービス提供方法。

【請求項 7】 請求項 1 乃至請求項 6 のいずれか記載の情報処理サービス提供方法において、

前記情報処理サービスに対して複数の付加的機能が連携して動作することを特徴とする情報処理サービスの提供方法。

【請求項 8】 請求項 1 乃至請求項 7 のいずれか記載の情報処理サービス提供方法において、

前記付加的な情報処理サービスについて、使用に応じた費用を計算することを特徴とする情報処理サービス提供方法。

【請求項 9】 コンピュータに、

ネットワークを経由して提供される情報処理サービスに対する付加的な情報処理サービスを依頼するイベントを検出する機能、

前記イベントが検出された場合に、前記付加的な情報処理サービスを提供する付加的機能の呼び出しを定義したライブラリを参照する機能、

前記ライブラリの参照結果に基づいて、前記付加的な情報処理サービスを提供する付加的機能に実行を依頼する機能
を実現させるためのプログラム。

【請求項 10】 コンピュータに、

ネットワークを経由して提供される情報処理サービスの運用状態を示すデータを記録する機能、

前記情報処理サービスに対する付加的な情報処理サービスを提供する付加的機能に対して、記録された前記運用状態を示すデータを提供する機能を実現させるためのプログラム。

【請求項 1 1】 コンピュータに、

ネットワークを経由して提供される情報処理サービスのリクエストを取得する機能、

前記リクエストに応じて、前記情報処理サービスに対する付加的な情報処理サービスを提供する付加的機能に実行を依頼する機能を実現させるためのプログラム。

【請求項 1 2】 コンピュータに、

ネットワークを経由して提供される情報処理サービスの処理に基づいて発生する前記情報処理サービスに対する付加的な情報処理サービスのリクエストを取得する機能、

前記リクエストに応じて、前記付加的な情報処理サービスを提供する付加的機能に実行を依頼する機能を実現させるためのプログラム。

【請求項 1 3】 請求項 1 1 又は請求項 1 2 記載のプログラムにおいて、

前記付加的な情報処理サービスを提供する付加的機能に実行を依頼するために、前記リクエストと前記付加的機能とを対応付けたテーブルを参照し、前記リクエストに対応する前記付加的機能を特定することを特徴とするプログラム。

【請求項 1 4】 情報処理サービスを提供するシステムにおいて、

ネットワークを経由して提供される情報処理サービスに対する付加的な情報処理サービスを依頼するイベントを検出するイベント検出手段と、

前記イベント検出手段によってイベントが検出された場合に、前記付加的な情報処理サービスを提供する付加的機能の呼び出しを定義したライブラリを参照する参照手段と、

前記参照手段によるライブラリの参照結果に基づいて、前記付加的な情報処理サービスを提供する付加的機能に実行を依頼する依頼手段とを具備したことを特徴とする情報処理サービス提供システム。

【請求項 1 5】 情報処理サービスを提供するシステムにおいて、

ネットワークを経由して提供される情報処理サービスの運用状態を示すデータを記録する記録手段と、

前記情報処理サービスに対する付加的な情報処理サービスを提供する付加的機能に対して、前記記録手段によって記録された運用状態を示すデータを提供する出力手段と

を具備したことを特徴とする情報処理サービス提供システム。

【請求項 1 6】 情報処理サービスを提供するシステムにおいて、

ネットワークを経由して提供される情報処理サービスのリクエストを取得するリクエスト取得手段と、

前記リクエスト取得手段によって取得されたリクエストに応じて、前記情報処理サービスに対する付加的な情報処理サービスを提供する付加的機能に実行を依頼する依頼手段と

を具備したことを特徴とする情報処理サービス提供システム。

【請求項 1 7】 情報処理サービスを提供するシステムにおいて、

ネットワークを経由して提供される情報処理サービスの処理に基づいて発生する前記情報処理サービスに対する付加的な情報処理サービスのリクエストを取得するリクエスト取得手段と、

前記リクエスト取得手段によって取得されたリクエストに応じて、前記付加的な情報処理サービスを提供する付加的機能に実行を依頼する依頼手段と

を具備したことを特徴とする情報処理サービス提供システム。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、情報処理サービス提供者を支援する情報処理サービス提供方法及びプログラム並びにシステムに関する。

【0 0 0 2】

【従来の技術】

サービス提供者とは、例えばビジネスに使用される業務ソフトウェアなどのア

アプリケーション・ソフトウェア（以下、アプリケーションという）の機能、又はその機能を提供する事業者である。サービス提供者は、アプリケーションによって得られた結果を、ユーザ又はシステムなどの依頼側に出力する。

【0003】

A S P（Application Service Provider）は、サービス提供者の一例である。A S Pは、例えば統合業務パッケージなどの大規模な業務システムの機能をレンタルの対象とする。

【0004】

また、A S Pは、文書編集ソフトウェア、表計算ソフトウェアなどのような日常的に使用されるアプリケーションもレンタルの対象とする。

【0005】

ユーザは、自己の端末にアプリケーションをインストールしなくても、A S Pのレンタルアプリケーションを使用し、作業可能である。

【0006】

これにより、ユーザ側では、企業の情報システム部門にとって負担となっていたアプリケーションのインストール、管理、アップグレードなどの費用、労力を軽減できる。

【0007】

一般的に、ユーザは、Webブラウザを使用してA S Pにアクセスする。しかしながら、Webブラウザを使用することなくA S Pにアクセスする場合もある。例えば、ユーザは、携帯電話専用のブラウザを使用してA S Pにアクセスする。また、ユーザは、A S Pから期限付きでレンタルされるソフトウェアを自己のハードディスクにダウンロードし、使用する。

【0008】

サービス提供者は、サービス料を得るために、ユーザの使用量に応じて課金する機能及びユーザ情報の管理機能を具備する必要がある。したがって、サービス提供者は、ユーザに使用されるアプリケーションと連携動作する課金管理機能及びユーザ管理機能を開発し、運用する必要がある。

【0009】

サービス提供者は、サーバのダウンや異なるユーザ間での情報漏洩を防止する必要がある。そのため、一般的に、サービス提供者は、サービスを提供する目的に特化した施設及び運用形態を提供するデータセンタ又はインターネットデータセンタ（以下、i D Cという）にサーバを設置し、サーバダウンや情報漏洩を防止する。

【 0 0 1 0 】

【発明が解決しようとする課題】

データセンタ又はi D Cは、物理的なインフラ及び運用管理をサービス提供者に使用させる。

【 0 0 1 1 】

しかしながら、サービス提供者がサービスを提供するために必要な機能の中には、データセンタ又はi D Cを使用するのみでは実現されない機能がある。

【 0 0 1 2 】

例えば、データセンタ又はi D Cを使用するのみでは、アプリケーションの使用量に応じた課金管理の機能、請求書作成支援機能、請求書発送を支援する機能、ユーザからの問い合わせに対して返答する機能等が十分に実現されない。

【 0 0 1 3 】

サービス提供者の中には、サービスを提供するために必要な全ての機能を自己で開発又は運用する技術力を持たない者もいる。このように、技術力が十分でないサービス提供者が上記機能を含むサービスを提供するのは困難であり、サービス提供の障害となる。

【 0 0 1 4 】

本発明は、以上のような実情に鑑みてなされたもので、サービス提供者、例えばA S Pを支援するための情報処理サービス提供方法及びプログラム並びにシステムを提供することを目的とする。

【 0 0 1 5 】

【課題を解決するための手段】

本発明を実現するにあたって講じた具体的手段について以下に説明する。

【 0 0 1 6 】

第1の発明では、コンピュータによる情報処理サービス提供方法において、ネットワークを経由して提供される情報処理サービスに対する付加的な情報処理サービスを依頼するイベントを検出し、イベントが検出された場合に、付加的な情報処理サービスを提供する付加的機能の呼び出しを定義したライブラリを参照し、ライブラリの参照結果に基づいて、付加的な情報処理サービスを提供する付加的機能に実行を依頼する。

【0017】

これにより、情報処理サービスと付加的な情報処理サービスとを連携させることができる。

【0018】

したがって、あるサービス提供者のみでは十分なサービスを提供できない場合であっても、このサービス提供者は、他のサービス提供者によって提供される付加的な情報処理サービスを使用し、サービスの価値を高めることができる。

【0019】

また、サービス提供者は、付加的な情報処理サービスを他のサービス提供者に依頼することで、自己が管理する機能を削減でき、管理業務を簡易化できる。

【0020】

また、iDC業者は、付加的な情報処理サービスを提供することにより、自データセンタの付加価値を高めることができる。

【0021】

なお、情報処理サービスと付加的な情報処理サービスとが、別個のハードウェア上で実行される場合には、ライブラリはネットワーク経由で使用可能としてもよい。

【0022】

付加的機能は、例えばプログラム、コンピュータ・ハードウェア、オブジェクト、モジュール、プロセスなどにより実現される。

【0023】

また、付加的機能に対する実行の依頼方法としては、例えば付加的機能にリクエストを出力する手法、付加的機能にデータを提供する手法、付加的機能の処理

を呼び出す手法などがある。

【 0 0 2 4 】

第 2 の発明では、コンピュータによる情報処理サービス提供方法において、ネットワークを経由して提供される情報処理サービスの運用状態を示すデータを記録し、情報処理サービスに対する付加的な情報処理サービスを提供する付加的機能に対して、記録された運用状態を示すデータを提供する。

【 0 0 2 5 】

これにより、情報処理サービスの運用状態を示すデータを使用する情報処理サービス（例えば、運用管理サービス、ユーザ管理サービス、課金管理サービスなど）を、付加的な情報処理サービスとして使用できる。

【 0 0 2 6 】

第 3 の発明では、コンピュータによる情報処理サービス提供方法において、ネットワークを経由して提供される情報処理サービスのリクエストを取得し、リクエストに応じて、情報処理サービスに対する付加的な情報処理サービスを提供する付加的機能に実行を依頼する。

【 0 0 2 7 】

この第 3 の発明では、情報処理サービスの依頼側から情報処理サービスの提供側に通知されるリクエストが取得され、この取得されたリクエストに対応する付加的な情報処理サービスが依頼される。

【 0 0 2 8 】

第 4 の発明では、コンピュータによる情報処理サービス提供方法において、ネットワークを経由して提供される情報処理サービスの処理に基づいて発生する情報処理サービスに対する付加的な情報処理サービスのリクエストを取得し、リクエストに応じて、付加的な情報処理サービスを提供する付加的機能に実行を依頼する。

【 0 0 2 9 】

この第 4 の発明では、情報処理サービスの処理実行により付加的な情報処理サービスのリクエストが発生した場合に、情報処理サービスの実行側から付加サービスの提供側に実行が依頼される。

【 0 0 3 0 】

これにより、例えば情報処理サービスの実行中又は実行後に必要となる課金処理を、課金管理を行う付加的な情報処理サービスにより行わせることができる。

【 0 0 3 1 】

なお、上記第 3 又は第 4 の発明において、付加的な情報処理サービスを提供する付加的機能に実行を依頼するために、リクエストと付加的機能とを対応付けたテーブルを参照し、リクエストに対応する付加的機能を特定するとしてもよい。

【 0 0 3 2 】

このようなテーブルを使用することで、リクエストに応じて適切な付加的機能を特定し、実行を依頼できる。

【 0 0 3 3 】

また、上記各発明において、付加的機能は、複数の情報処理サービスに共同で使用されてもよい。

【 0 0 3 4 】

これにより、複数のサービス提供者間で例えばユーザ管理や課金管理などの付加的な情報処理サービスを統一化できる。また、複数のサービス提供者についてのシングルサインオン環境をユーザに提供できる。また、ユーザに対する課金を統合し、ユーザの支払作業を簡略化できる。

【 0 0 3 5 】

また、上記各発明において、情報処理サービスに対して複数の付加的機能が連携して動作するとしてもよい。

【 0 0 3 6 】

これにより、例えば付加的な情報処理サービスを提供するサービス業者がさらに他の付加的な情報処理サービスを提供するサービス業者に対してサービスを依頼することができる。したがって、付加的な情報処理サービス群を連携させて一つの情報処理サービスを実現できる。

【 0 0 3 7 】

また、上記各発明において、付加的な情報処理サービスについて、使用に応じた費用を計算するとしてもよい。

【 0 0 3 8 】

これにより、付加的な情報処理サービスに対する料金を得ることができる。

【 0 0 3 9 】

なお、上記各発明における情報処理サービス提供方法は、コンピュータに読み込まれたプログラムにより実現させてもよい。また、このプログラムは、コンピュータ読み取り可能な記録媒体に記録されてもよい。

【 0 0 4 0 】

このプログラム又はこのプログラムを記録した記録媒体を用いることによって、コンピュータシステム、サーバマシン又はクライアントマシン等のコンピュータに、上述した発明を実施する機能を実現させることができる。

【 0 0 4 1 】

また、上記各発明と同様の動作を実現する手段を情報処理サービス提供システムに具備することにより、同様の作用効果を得ることができる。

【 0 0 4 2 】

【発明の実施の形態】

以下、図面を参照しながら本発明の実施の形態について説明する。なお、以下に示す各図において、同一の部分については同一の符号を付する。

【 0 0 4 3 】

以下の説明においては、主にユーザがWebブラウザなどを操作してリクエストを発信する場合について説明する。しかしながら、これに限定されるものではなく、プログラム、コンピュータ、コンピュータシステム、オブジェクト、モジュール、プロセスなどがリクエストを発信してもよい。

【 0 0 4 4 】

(第1の実施の形態)

本実施の形態においては、情報処理サービスに対する付加的な情報処理サービス（以下、「付加サービス」という）を提供する情報処理サービス提供システムについて説明する。

【 0 0 4 5 】

なお、本実施の形態においては、ASP業務を行うための情報技術サービスが

情報処理サービスである場合を例として説明する。ASPは、サービスの依頼側にアプリケーションサービスを提供する。付加サービスの一例としては、ASPに対する運用管理サービスがある。

【0046】

図1は、本実施の形態に係る情報処理サービス提供システムの構成の一例を示すブロック図である。

【0047】

情報処理サービス提供システムのユーザ（エンドユーザ及びユーザ企業など）2は、自己の操作するクライアント3からインターネット4を経由してASPサーバ5a又はASPサーバ5bにリクエストを提供する。このリクエストに基づいてASPサーバ5aのアプリケーション6a又はASPサーバ5bのアプリケーション6bは稼動する。その結果、ユーザ2は、ASPサービスを受ける。

【0048】

ASPサーバ5a、5bのイベント検出機能7は、付加サービスを依頼するイベントの発生を検出する。

【0049】

例えば、ASPサーバ5a、5bは、新規のユーザにアクセスされた場合、この新規ユーザを管理するために新規ユーザの登録を必要とする。この場合、イベント検出機能7は、新規のユーザからのアクセスを検出する。その他にも、イベント検出機能7は、ユーザの使用に応じてそのユーザに対して課金するためのイベント、またASPサーバ5a、5bの障害発生を示すイベントを検出する。

【0050】

ライブラリ8は、付加サービス11a～11cを提供する機能の呼び出しに使用されるデータ又はプログラムを格納している。ライブラリ8は、ASPサーバ5a、5bから付加サービスを提供する機能を呼び出すためのAPI（Application Program Interface）を具備する。なお、付加サービスを提供する機能は、例えばプログラム、プロセス、モジュール、アプリケーション、コンピュータ、コンピュータシステムなどで実現される。

【0051】

ライブラリ 8 は、参照機能 9 を実現するハードウェアと同一のハードウェアに具備されない場合もある。そのため、参照機能 9 は、ネットワーク経由でライブラリ 8 を参照可能とする。

【0052】

参照機能 9 は、イベント検出機能 7 でイベントが検出された場合にライブラリ 8 を参照する。

【0053】

依頼機能 10 は、参照機能 9 によるライブラリ 8 の参照結果を使用し、イベント検出機能 7 で検出されたイベントの種別に応じた機能呼び出す。

【0054】

付加サービス 11 a ～ 11 c は、そのサービスを提供する機能に実行が依頼されると実現される。

【0055】

上記図 1 では、付加サーバ 12 a によってユーザ管理サービス 11 a 及び課金管理サービス 11 b が提供される。また、付加サーバ 12 b によってセキュリティサービス 11 c が提供される。

【0056】

ASPサーバ 5 a、5 b と付加サーバ 12 a、12 b とは、ネットワーク 4 a を介して接続されている。

【0057】

なお、本実施の形態において、イベント検出機能 7、参照機能 9、ライブラリ 8、依頼機能 10 は ASPサーバ 5 a、5 b 側に配置されているが、例えば付加サーバ 12 a、12 b 側に配置されてもよい。付加サーバ 12 a、12 b 側にイベント検出機能 7 が配置された場合、このイベント検出機能 7 はネットワーク 4 a を経由して ASPサーバ 5 a、5 b で発生するイベントを検出する。

【0058】

また、本実施の形態においては、ASPサーバ 5 a、5 b と付加サーバ 12 a、12 b とが異なるハードウェア上で稼動する場合を例として説明している。しかしながら、ASPサーバ 5 a、5 b と付加サーバ 12 a、12 b は同一のハー

ドウェア上で稼動してもよい。

【0059】

図2は、本実施の形態に係る情報処理サービス提供システムの構成の変形例を示すブロック図である。

【0060】

ASPサーバ5cの記録機能13は、ASPサーバ5cの運用状態データをデータベース14に記録する。運用状態データには、例えばユーザ2のアプリケーション6cの使用回数、使用時間、ASPサーバ5cの動作履歴情報などがある。

【0061】

ASPサーバ5cの依頼機能15は、付加サービスを提供する機能に対してデータベース14に記録されている運用状態データを出力する。

【0062】

付加サーバ12aは、運用状態データを入力すると、ユーザ管理サービス11a又は課金管理サービス11bを提供し、付加サーバ12bは、運用状態データを入力すると、セキュリティサービス11cを提供する。

【0063】

以上説明した情報処理サービス提供システムにおいては、ASPのアプリケーション6a、6bと付加サービス11a～11cを提供する機能とが連携して動作する。

【0064】

したがって、例えばiDC業者が各種付加サービスを提供すれば、このiDC業者は自己のデータセンタの付加価値を高めて事業を行うことができる。

【0065】

また、ユーザ2が複数のASPサービスを受ける場合、使用される複数のASPが共通の付加サービスを使用することにより、このユーザ2に対するユーザ管理や課金管理を複数のASP間で統一化することができる。

【0066】

したがって、ユーザ2は、複数のASPに対するシングルサインオン環境を使

用可能であり、複数のASPへの支払いを統合できる。

【0067】

また、ASPは、ASPアプリケーションとしての仕様で作成されていないアプリケーションに対して、ASPアプリケーションに必要な機能を付加する付加サービスを受け、容易にASPアプリケーションとして使用することができる。

【0068】

したがって、ASPは各種アプリケーションを容易にレンタルすることができる。また、ASPは、自己の提供するサービスを充実させてユーザの使用を促進させることができる。

【0069】

なお、本実施の形態に係る情報処理サービス提供システムの各種機能7～10、13、15は、記録媒体16に記録されたプログラムをコンピュータが読み取ることによって実現可能である。

【0070】

(第2の実施の形態)

本実施の形態においては、ASPサービスと付加サービスとを連携させる情報処理サービス提供システムについて詳細に説明する。

【0071】

図3は、本実施の形態に係る情報処理サービス提供システムの連携状態の一例を示すブロック図である。

【0072】

ユーザ2は、ASP18のアプリケーションによるASPサービス18a～18gを使用する。

【0073】

ASPポータル17は、ユーザ2とASP18とのマッチメイク、ASP18に関するマーケティングを行う。さらに、ASPポータル17は、ASP間の連携とそのための情報処理サービス、ユーザに対するASP導入のためのコンサルティング、BPR (Business Process Re-engineering) 支援サービスを提供する。

【0074】

ASP18は、ASPアプリケーションによるASPサービス18a～18gをユーザ2に提供する。

【0075】

付加サービス業者19は、複数のASPに対する共通の情報処理サービス、ASPの運用管理サービスなどの付加サービス11a～11jを提供する。

【0076】

付加サービスの一種である共通運用管理サービス11jは、付加サービス業者19が他の付加サービス業者の提供する付加サービス又は他のASPの提供するASPサービスなどを受けるための管理を行う。

【0077】

iDC業者20は、物理インフラ及び運用管理サービスを提供する。また、iDC業者20は、ホスティング(サーバ貸し)、ハウジング(場所貸し)を行う。

【0078】

ASP18によって提供されるASPサービス18a～18gは、必要に応じて付加サービス業者19によって提供される付加サービス11a～11j及びiDC業者20のiDC21を使用する。

【0079】

iDC21上には、付加サービス11a～11jを提供するためのアプリケーションプラットフォーム22が具備されている。

【0080】

図4は、本実施の形態に係る情報処理サービス提供システムの構成の一例を示すブロック図である。

【0081】

本実施の形態では、付加サービス11a～11jをその役割に基づいて、ベーシックサービス、コミュニケーションサービス、ユーティリティサービス、バックエンドサービスの4種に分類する。

【0082】

ベーシックサービスは、ASPサービス18a～18gを提供するために必要

な基本機能を提供するサービスである。ベーシックサービスの例としては、ユーザ管理サービス 1 1 a、課金管理サービス 1 1 b、セキュリティサービス 1 1 c、SLM (Service Level Management) サービス 1 1 d などがある。なお、SLMとは、ASPがユーザに対してサービスの質（稼働率やレスポンス）を保証するためのシステム管理と、このシステム管理機能を使用したサービスレベル保証をいう。

【 0 0 8 3 】

ユーティリティサービスは、ASPサービス 1 8 a ～ 1 8 g を容易に開始するためのサービスである。ASP 1 8 は、自己のサービスに対してユーティリティサービスを必要に応じて組みこむことで、ASP 1 8 自身で提供することが困難なサービスを簡単に提供可能となる。ユーティリティサービスの例としては、情報分析サービス、情報提供サービス、印刷サービスなどがある。

【 0 0 8 4 】

コミュニケーションサービスは、ユーザ 2 と ASP 1 8 との間のコミュニケーションを確立するためのサービスである。コミュニケーションサービスの例としては、ポータルサービス、ユーザ 2 の電子メールの使用を可能とする電子メールサービス 1 1 k などがある。

【 0 0 8 5 】

バックエンドサービスは、ASPサービス 1 8 a ～ 1 8 g の運用のためにバックエンドで処理を実行するサービスである。バックエンドサービスの例としては、EDI (Electronic Data Interchange) / EAI (Enterprise Application Integration) サービス、ジョブコントロールサービス 1 1 i などがある。

【 0 0 8 6 】

なお、EDIとは、商取引に関する情報を標準的な書式に統一して、企業間で電子的に交換する仕組みである。この仕組みでは、受発注や見積もり、決済、出入荷などに関わるデータを予め定められた形式にしたがって電子化し、専用線やVANなどのネットワークを通じて送受信する。

【 0 0 8 7 】

また、EAIとは、企業内で業務に使用される複数のコンピュータシステムを

有機的に連携させ、データやプロセスの効率的に統合し、またこの統合を支援する技術及びソフトウェアである。

【 0 0 8 8 】

さらに、本実施の形態においては、A S P サービス 1 8 a ～ 1 8 g と付加サービス 1 1 a ～ 1 1 j との間の連携方法に基づいて、付加サービス 1 1 a ～ 1 1 j をユーティリティ連携サービス、ベーシック連携サービス、バックエンド連携サービスの 3 種に分類する。

【 0 0 8 9 】

ユーティリティ連携サービスは、ユーザ 2 又は A S P 1 8 から必要に応じて呼び出されるサービスであり、主に上記のユーティリティサービス及びコミュニケーションサービスが含まれる。ユーティリティ連携サービスの呼び出しは、ユーティリティ連携サービスにアクセスするルーチンを A S P アプリケーションの内部に組みこむことで実現する。

【 0 0 9 0 】

ベーシック連携サービスは、ユーザ 2 が A S P 1 8 の A S P アプリケーションにアクセスする際に介在するサービスである。ユーザ 2 の認証を行うユーザ管理サービス 1 1 a、A S P 1 8 によるユーザ 2 に対する課金管理サービス 1 1 b などのような上記ベーシックサービスは、ベーシック連携サービスに該当する。ベーシック連携サービスは、ユーザ 2 の操作するクライアントから A S P 1 8 に対するサービス実行リクエストが発生する度に自動的に起動される。

【 0 0 9 1 】

加えて、このベーシック連携サービスには、A S P サービス 1 8 a ～ 1 8 g から付加サービス 1 1 a ～ 1 1 j を使用する際に介在するサービスも含まれる。付加サービス業者 1 9 から A S P 1 8 に対する課金管理などのようなベーシックサービスは、ベーシック連携サービスに該当する。ベーシック連携サービスは、A S P サービス 1 8 a ～ 1 8 g が付加サービス 1 1 a ～ 1 1 j を呼び出す際に自動的に起動される。

【 0 0 9 2 】

バックエンド連携サービスは、A S P 1 8 自体の稼動状態を監視・制御するサ

ービスである。上記バックエンドサービスは、このバックエンド連携サービスに該当する。バックエンド連携サービスは、付加サービス業者 1 9 側から A S P 1 8 に対して処理の起動・停止などの操作を行う。

【0093】

このようなユーティリティ連携サービス、ベーシック連携サービス、バックエンド連携サービスと A S P サービス 1 8 a ～ 1 8 g との間の連携を実現するために必要な 3 種類のインタフェースについて説明する。

【0094】

各種 A S P サービス 1 8 a ～ 1 8 g 及び付加サービス 1 1 a ～ 1 1 j は、それぞれに対応するプログラム又はハードウェアにリクエストが出力された場合、またはそれぞれに対応するプログラム又はハードウェアが呼び出された場合に実行される。

【0095】

ユーティリティ連携サービスと A S P サービス 1 8 a ～ 1 8 g の連携、ベーシック連携サービスと A S P サービス 1 8 a ～ 1 8 g の連携、バックエンド連携サービスと A S P サービス 1 8 a ～ 1 8 g との連携は、以下のアーキテクチャで実現される。

【0096】

ユーティリティ連携サービスと A S P サービス 1 8 a ～ 1 8 g との間の連携を可能とするために、情報処理サービス提供システムは、A S P サービス 1 8 a ～ 1 8 g からユーティリティ連携サービスを呼び出すための A P I を含むライブラリ 8 を具備する。そして、A S P サービス 1 8 a ～ 1 8 g を提供する時点で、ライブラリ 8 を使用して A S P サービス 1 8 a ～ 1 8 g 側から付加サービス 1 1 a ～ 1 1 j を呼び出す。これにより、ユーティリティ連携サービスと A S P サービス 1 8 a ～ 1 8 g との間の連携が可能となる。

【0097】

なお、呼び出し側の A S P サービス 1 8 a ～ 1 8 g と呼び出される側のユーティリティ連携サービスとが別のハードウェア上で稼動する場合があるため、本実施の形態においても、上記第 1 の実施の形態と同様に、ライブラリ 8 をネットワ

ーク経由で使用可能とすることが望ましい。

【0098】

ベーシック連携サービスとASPサービス18a～18gとの間の連携を可能とするために、情報処理サービス提供システムは、ユーザ2の操作するクライアントからのリクエストを受け付けるネットワークサーバ（例えばWebサーバなどのプロセス）23の内部に、ベーシック連携サービスの実行を依頼するユーザ側リクエスト取得機能24を具備することで連携を可能とする。

【0099】

この場合の処理の流れは、まずネットワークサーバ23がユーザ2からASPサービス18a～18gのリクエストを受け取る。ユーザ側リクエスト取得機能24のリクエスト取得機能24aはこのユーザ2からASPサービス18a～18gのリクエストを取得し、依頼機能24bはベーシック連携サービスを提供する機能に実行を依頼する。これにより、一旦ベーシック連携サービスを提供する機能に制御が移る。そして、ユーザ認証などの必要な機能を実行し、ネットワークサーバ23は、ASPサービス18a～18gにリクエストを発する。これにより、ASP18a～18gを提供するアプリケーションに制御が移る。

【0100】

また、情報処理サービス提供システムは、ASPサービス18a～18gを提供するアプリケーションからユーティリティ連携サービスに対するリクエストをリクエスト取得機能25aで取得し、依頼機能25bでベーシック連携サービスを提供する機能に処理実行を依頼するアプリケーション側リクエスト取得機能25を具備する。

【0101】

バックエンド連携サービスとASPサービス18a～18gとの間の連携を可能とするために、情報処理サービス提供システムは、ASPサービス18a～18gを提供するアプリケーションのプロセス状態を定常的に監視し、その状態に応じてバックエンド連携サービスからASPサービス18a～18gを実現するアプリケーションのプロセスを操作し、連携を実現する。なお、ASPサービス18a～18gが独立したプロセスでなく、アプリケーションサーバ上で動作す

るスレッドの場合には、アプリケーションサーバの持つ監視機能をバックエンド連携サービスから呼び出す。

【0102】

上記のような連携のためのアーキテクチャを実現するためのリクエスト取得機能について以下で詳細に説明する。

【0103】

上述したように、ユーザ側リクエスト取得機能24とアプリケーション側リクエスト取得機能25とを含む2種類のリクエスト取得機能は、リクエストを取得し、このリクエストに対応するサービス又は付加サービスの提供を依頼する。

【0104】

図5は、ユーザ側リクエスト取得機能24による連携状態の一例を示すブロック図である。

【0105】

ユーザ側リクエスト取得機能24は、ユーザ2からASPサービス18a～18gに対するリクエストを取得し、ベーシック連携サービスを呼び出す。この呼び出しにより、各種ASPサービス18a～18gを使用するユーザ2が識別され、ユーザ認証及び課金処理が実行される。ASP18a～18gを提供するアプリケーションは、ユーザ2からのリクエストを受け付け、結果を返す。

【0106】

すなわち、このユーザ側リクエスト取得機能24は、ユーザ2からのASPサービス18a～18gの使用に介在し、ユーザ認証や与信・課金などのユーザ管理機能の実行を依頼する。

【0107】

図6は、アプリケーション側リクエスト取得機能25による連携状態の一例を示すブロック図である。

【0108】

アプリケーション側リクエスト取得機能25は、ASPサービス18a～18gを提供するアプリケーションからユーティリティ連携サービスを使用する旨のリクエストを取得し、ベーシック連携サービスを呼び出す。この呼び出しにより

、ASP18が付加サービス11a～11jを使用した量に応じて、ASP18に対して適正に課金することが可能になる。ユーティリティ連携サービスを提供する機能は、ASP18からのリクエストを受け付け、結果を返す。

【0109】

すなわち、このアプリケーション側リクエスト取得機能25は、ASP18による付加サービス11a～11jの使用に介在し、認証や与信・課金などの管理処理の実行を依頼する。

【0110】

以下に、リクエスト取得機能24、25の実現方法を説明する。なお、以下においてはユーザ2からWebサーバを経由してASP18にリクエストが取得される場合について説明する。しかしながら、リクエスト取得機能24、25は、ユーザ2からのアクセスを受け付けるあらゆる種類のネットワークサーバ23に同様に適用可能である。以下で説明する付加サービスの例としては、iDC21を使用する1以上のASPを使用するユーザの情報を統合して管理するサービス、1以上のASPの課金情報を統合して管理するサービス、1以上のASPに関する稼動状態を一元管理するサービス、サービスを提供する1以上の機能の稼動状態を一元管理するサービスなどがある。

【0111】

リクエスト取得機能24、25は、ASPサービスを提供する機能呼び出すリクエスト又はユーティリティ連携サービスを提供する機能呼び出すリクエストを取得し、ベーシック連携サービスを提供する機能にリクエストに関わる情報を出力する。これにより、上記の付加サービスが実行される。ユーザ2はWebサーバを経由してASPアプリケーションを使用する。

【0112】

ベーシック連携サービス31には、ユーザ管理サービス11a、課金管理サービス（予信・課金サービス）11bなどがある。ユーティリティ連携サービス35には、電子メールサービス11kなどがある。

【0113】

ユーザ管理サービス11aは、指定されたパスワードによりユーザを認証し、

またユーザに要求されたデータへのアクセス権を確認するサービスである。

【0114】

課金管理サービス11bは、指定されたユーザの予信状態（使用料の上限に達しているか否か等）を調査し、課金用の情報を保持して月毎に集計するサービスである。

【0115】

電子メールサービス11kは、大量の送付先に広告用のメールを送り、その中に各送付先に特化した情報（宛先ユーザの名前など）を含めるサービスである。

【0116】

図7は、リクエスト取得機能24、25を含む情報処理サービス提供システムのアーキテクチャの一例を示すブロック図である。

【0117】

また、図8は、情報処理サービス提供システムのアーキテクチャと具体的なサービスとの関係の一例を示すブロック図である。

【0118】

ユーザ側リクエスト取得機能24とアプリケーション側リクエスト取得機能25とは、機能的に似た点が多いため共通部分を部品化できる。共通化した部品をリクエスト取得フレームワーク26とする。

【0119】

リクエスト取得フレームワーク26は例えばオブジェクト指向プログラミング言語で開発される。リクエスト取得フレームワーク26の使用する対応テーブルは、ファイル等に記録されて管理される。この対応テーブルを使用することで、リクエストがどのサービスに対応するか特定され、リクエストの種類によって起動するベーシック連携サービスの機能を切り換え可能となる。

【0120】

ユーザ側リクエスト取得機能24は、Webサーバ28又はその前段のProxyサーバ（以下、「Gateサーバ」という）27に含まれる機能とする。すなわち、Gateサーバ27は、ユーザ側リクエスト取得機能24を実現している。

【 0 1 2 1 】

ユーザ 2 は W e b ブラウザ 2 9 から W e b サーバ 2 8 にアクセスする。すると、G a t e サーバ 2 7 のユーザ側リクエスト取得機能 2 4 は、W e b ブラウザ 2 9 から W e b サーバ 2 8 へのリクエストを取得する。そして、ユーザ側リクエスト取得機能 2 4 は、ベーシックサービスアダプタ 3 0 （例えばユーザ管理アダプタ 3 0 a、課金管理アダプタ 3 0 b）を用いて、ベーシック連携サービス 3 1 （例えばユーザ管理サービス 3 1 a、課金管理サービス 3 1 b）を提供する機能（例えばプロセス）に実行を依頼する。

【 0 1 2 2 】

この結果、アクセスしたユーザ 2 が特定され、その課金情報が蓄えられ、アクセスしたユーザ 2 がアプリケーションサーバ 3 2 上の A S P アプリケーション 3 3 を使用可能か否か判断される。

【 0 1 2 3 】

ユーザ 2 に A S P アプリケーション 3 3 の使用が許可されると、ユーザ 2 は A S P アプリケーション 3 3 を使用する。さらに、この A S P アプリケーション 3 3 に加えてユーザ 2 がユーティリティ連携サービス 3 5 （例えば電子メールサービス 3 5 a）を使用する場合、アプリケーション側リクエスト取得機能 2 5 は、A S P アプリケーション 3 3 からユーティリティ連携サービス 3 5 を提供する機能（例えばプロセス）へのリクエストを取得する。

【 0 1 2 4 】

そして、アプリケーション側リクエスト取得機能 2 5 は、ベーシックサービスアダプタ 3 0 を用いて、ベーシック連携サービス 3 1 を提供する機能に実行を依頼する。

【 0 1 2 5 】

この結果、ユーザ 2 が特定され、その課金情報が蓄えられ、ユーザ 2 がユーティリティ連携サービス 3 5 を使用可能か否か判断される。

【 0 1 2 6 】

ユーティリティ連携サービス 3 5 の使用が許可されると、A S P アプリケーション 3 3 はユーティリティサービスアダプタ 3 4 （例えば電子メールサービスア

ダブタ 3 4 a) を使用し、ユーティリティ連携サービス 3 5 を提供する機能に実行を依頼する。

【 0 1 2 7 】

以上のような連携により、ユーザ管理サービス 1 1 a、課金管理サービス 1 1 b が A S P 1 8 に提供される。

【 0 1 2 8 】

この結果、A S P 1 8 は、本来サービスとして備えていない電子メールサービス 1 1 k をユーザ 2 に提供でき、自分のサービスに新たな価値を加えることができる。

【 0 1 2 9 】

付加サービス業者 1 9 は、電子メールサービス 1 1 k を A S P 1 8 に提供した場合に、その使用に応じた課金情報を蓄え、A S P 1 8 に請求することができる。

【 0 1 3 0 】

図 9 は、リクエスト取得機能 2 4、2 5 の構造の一例を示すクラス図である。この図は設計表記法の一つである UML (Unified Modeling Language) の形式で記述されている。

【 0 1 3 1 】

また、図 1 0 は、リクエスト取得機能 2 4、2 5 の構造と具体的なサービスとの関係の一例を示すクラス図である。

【 0 1 3 2 】

矩形及び 3 分割された矩形は、クラスを表している。矩形の中に記述された名前（例えば「G a t e サーバ」）はこのクラスの名前である。3 分割された矩形はクラスをより詳細に表記している。矩形の最上部の中身（例えば「取得ライブラリ」）はクラス名を、矩形の最下部の中身（例えば「前処理()」、「後処理()」）は、オブジェクト指向の場合にはメソッドに該当し、そのクラスが受け付けることのできる処理要求を示す。

【 0 1 3 3 】

破線の矢印は、根元側のクラスが先端側のクラスにアクセスして処理を呼び出

したり情報を得たりすることを表している。

【 0 1 3 4 】

クラス 3 0（例えばユーザ管理アダプタ 3 0 a、課金管理アダプタ 3 0 b）は、ベーシックサービスアダプタ規約 3 8 に準拠し、処理メソッドを持ち、取得ライブラリ 3 7 から呼び出される。

【 0 1 3 5 】

クラス 3 0 と取得ライブラリ 3 7 とは、ユーザ側リクエスト取得機能 2 4 とアプリケーション側リクエスト取得機能 2 5 との共通の機能である。

【 0 1 3 6 】

取得ライブラリ 3 7 は、リクエスト取得機能 2 4、2 5 の基礎になるクラスである。

【 0 1 3 7 】

G a t e サーバ 2 7 は、取得ライブラリ 3 7 を呼び出し、ユーザ側リクエスト取得機能 2 4 を実行する。

【 0 1 3 8 】

ユーティリティサービスアダプタ 3 4（例えば電子メールサービスアダプタ 3 4 a）は、ユーティリティサービスアダプタ規約 3 9 に準拠し、A S P アプリケーション 3 3 によってユーティリティ連携サービス 3 5（例えば電子メールサービス 3 5 a）が呼び出される場合に使用される。

【 0 1 3 9 】

ユーティリティサービスアダプタ 3 4 は、取得ライブラリ 3 7 を呼び出す。

【 0 1 4 0 】

リクエスト取得機能 2 4、2 5 を構成する各クラスの役割を以下に示す。

【 0 1 4 1 】

G a t e サーバ 2 7 は、クライアント 2 9 から A S P アプリケーション 3 3 に対するリクエストを仲介する P r o x y サーバである。G a t e サーバ 2 7 は、リクエストを転送する前と後の少なくとも一方で、取得ライブラリ 3 7 を呼び出す。

【 0 1 4 2 】

ASPアプリケーション33は、ASPサービスを提供する機能である。ASPアプリケーション33は、サービスの実行中にユーティリティ連携サービス35（例えば電子メールサービス35a）を使用する。

【0143】

取得ライブラリ37は、リクエスト取得機能の基礎となる。この取得ライブラリ37は、ASPアプリケーション33又はユーティリティ連携サービス35を提供する機能の実行前に呼び出される前処理メソッドと、実行後に呼び出される後処理メソッド、及びアダプタに渡す情報を受け取るための情報設定メソッドを含む。

【0144】

前処置メソッド及び後処理メソッドは、サービスアダプタ対応付け部40の内容にしたがって、ベーシックサービスアダプタ30を呼び出す。ベーシックサービスアダプタ30は、ベーシックサービスアダプタ規約38に準拠する。

【0145】

ベーシックサービスアダプタ規約38は、アダプタオブジェクトが準拠すべき規約である。アダプタオブジェクトは、リクエスト取得機能24、25によって起動されるベーシック連携サービス31を提供する機能を取得ライブラリ37から起動する。

【0146】

具体的には、ベーシックサービスアダプタ規約38では、処理メソッド及び情報設定メソッドを具備する旨が定義されている。この規約に準拠しているアダプタであれば、ベーシックサービスアダプタとして任意のアダプタを使用可能である。iDC業者20は、ベーシックサービスアダプタ規約38に準拠しているアダプタを用意することにより、ASP18に提供するベーシックサービスを随時変更できる。

【0147】

ベーシックサービスアダプタ30（例えば予信・課金アダプタ、ユーザ管理アダプタ）は、ベーシックサービス規約に準拠したクラスである。

【0148】

ベーシック連携サービス 31（例えばユーザ管理サービス 31a、課金管理サービス 31b）は、ユーザ認証、課金、実行状態ログの記録などを実行するサービスである。

【0149】

ユーティリティサービスアダプタ規約 39 は、ASP アプリケーション 33 から使用されるユーティリティサービスアダプタ 34 が準拠すべき規約である。

【0150】

ユーティリティサービスアダプタ 34（例えば電子メールサービスアダプタ 34a）は、ユーティリティ連携サービス 35 を提供する機能に ASP アプリケーション 33 からのリクエストを転送する役割を果たすクラスである。このクラスでは、転送の前後に取得ライブラリ 37 の前処理メソッド、後処理メソッドを呼び出す。

【0151】

ユーティリティ連携サービス 35（例えば電子メールサービス 35a）は、ASP アプリケーション 33 から使用されるサービスである。

【0152】

図 11 は、ユーザ側リクエスト取得機能 24 の前段動作の一例を示すシーケンス図である。この図 11 は、ユーザが ASP アプリケーション 33 を使用するまでの処理について示している。

【0153】

また、図 12 は、ユーザ側リクエスト取得機能 24 の後段動作の一例を示すシーケンス図である。この図 12 は、ユーザ 2 が ASP アプリケーション 33 の使用を終えた後の処理について示している。

【0154】

ユーザ 2 の Web ブラウザ 29 によって ASP アプリケーション 33 が使用される前において、ユーザ管理サービス 31a を提供する機能は、ユーザ 2 の認証チェックを行い、課金管理サービス 31b を提供する機能は、ユーザ 2 の ASP サービス使用料のチェックを行う。

【0155】

また、ユーザ 2 によって A S P アプリケーション 3 3 が使用された後において、課金管理サービスを提供する機能は、ユーザ 2 の課金情報を蓄積する。

【 0 1 5 6 】

シーケンス図中の実線の矢印はメソッドの呼出を、破線の矢印は呼び出された処理を終えて制御が戻ることを示している。両矢印には、必要に応じて呼び出されるメソッド名（実線の矢印の場合、引数は（）で囲まれて記述されている）と返される値が付されている。返される値も（）で囲まれて記述されている。取得ライブラリ 3 7 を実現する上で重要な働きを持つメソッドの呼び出しは、太線で強調している。

【 0 1 5 7 】

まず、図 1 1 に示すように、ユーザ 2 は、W e b ブラウザ 2 9 などのクライアントを用いて A S P アプリケーション 3 3 の使用を要求する。この時点でユーザ 2 は W e b サーバ 2 8 に未だアクセスしておらず、実際には G a t e サーバ 2 7 がこのリクエストを受け付けている。

【 0 1 5 8 】

次に、G a t e サーバ 2 7 は、ユーザ 2 によって指定された U R L に基づいて、依頼されているサービスが何か把握する検索処理を実行する。この検索処理は、サービス－U R L 対応付け部 4 1 に依頼する。

【 0 1 5 9 】

表 1 は、サービス－U R L 対応付け部 4 1 が使用する対応テーブルを例示している。

【 0 1 6 0 】

【表 1】

表 1

サービス名	U R L
ASP トップ	/ASP/ASP.jsp
課金情報閲覧	/ASP/accounting.jsp
付帯	/ASP/option.jsp
システム連携 1	/ASP/trans.jsp
システム連携 2	/servlet/asp.transSvr
XYZ	/XYZ/xyz.jsp

【0161】

この表 1 を参照することで、U R L から「X Y Z」などのようなサービス名を得ることができる。

【0162】

次に、G a t e サーバ 2 7 は、このようにして得たサービス名及びユーザ 2 が入力したユーザ名などの情報を取得ライブラリ 3 7 に渡す。サービス名、ユーザ名などの情報は、後に取得ライブラリ 3 7 から各アダプタに渡され、ベーシック連携サービスを提供する機能が処理を実行するための引数として使われる。

【0163】

表 2 は、ユーザ側リクエスト取得機能 2 4 がベーシック連携サービスを提供する機能に引数として渡す値の例である。

【0164】

【表 2】

表 2

項 目 名	値
利用者名	“ABCD”
パスワード	“1111111111”
呼出側サービス名	“XYZ”
呼出先サービス名	“Gate サーバ”
データサイズ	-
データ型	-
終了状態	-

【0165】

表3は、アプリケーション側リクエスト取得機能25がベーシック連携サービスを提供する機能に引数として渡す値の例である。

【0166】

【表3】

表 3

項 目 名	値
利用者名	“ASP33 管理者”
パスワード	“aaaaaa”
呼出側サービス名	“電子メール. メール送信”
呼出先サービス名	“XYZ”
データサイズ	-
データ型	-
終了状態	-

【0167】

次に、Gateサーバ27は、取得ライブラリ37の前処理メソッドを実行する。すると、取得ライブラリ37は、サービスアダプタ対応付け部40にアダプタ取得を依頼し、ASPアプリケーション33の前処理に必要なアダプタ群を取得する。

【0168】

表4は、サービスアダプタ対応付け部40が使用するテーブルを例示している。

【0169】

【表 4】

表 4

サ ー ビ ス	ア ダ プ タ	
XYZ. 前処理	ユーザ管理、課金管理	ユーザ側リクエスト 取得機能用
XYZ. 後処理	課金管理	
システム連携 1. 前処理	ユーザ管理	
システム連携 2. 後処理	課金管理、 付加サービス利用可	
...	...	
電子メール. メール送信. 前処理	ユーザ管理、課金管理	アプリケーション側 リクエスト 取得機能用
電子メール. メール送信. 後処理	課金管理	
電子メール. メール受信. 前処理	ユーザ管理、課金管理	
電子メール. メール受信. 後処理	課金管理、 付加サービス利用可	
...	...	

【0170】

具体的には、取得ライブラリ 37 は、既に表 2 のような情報を受け取っているため、「XYZ」をキーワードとして表 4 に対する検索を行う。

【0171】

これにより、取得ライブラリ 37 は、ASP アプリケーション 33 の前処理に必要なアダプタとしてユーザ管理アダプタ 30 a、課金管理アダプタ 30 b を選択する。

【0172】

次に、取得ライブラリ 37 は、選択したアダプタを使用して実際のサービスを提供する機能呼び出す。取得ライブラリ 37 は、先にユーザ管理サービス 31 a を提供する機能呼び出し、次に課金管理サービス 31 b を提供する機能呼び出す。具体的には、取得ライブラリ 37 は、先の処理で設定された表 2 の情報を選択されたアダプタに渡し、各アダプタは、自己の備えている処理メソッドを呼び出す。

【0173】

ユーザ管理アダプタ 30 a は、ユーザ管理サービス 31 a を提供する機能に特

化した操作メソッドを呼び出すとともに、ユーザ管理サービス 3 1 a を提供する機能にユーザ名とパスワードとを渡す。

【 0 1 7 4 】

すると、ユーザ管理アダプタ 3 0 a は、ユーザが正当であるか否かチェックした結果を、ユーザ管理サービス 3 1 a を提供する機能から受け取る。その結果は、さらにユーザ管理アダプタ 3 0 a から取得ライブラリ 3 7 に返される。

【 0 1 7 5 】

チェック結果に問題がなければ、同様に課金管理アダプタ 3 0 b は課金管理サービス 3 1 b のメソッドを呼び出す。これにより、課金管理サービス 3 1 b は、例えばこのユーザの使用料が契約上の上限値を上回っているか否かチェックする予信管理を行う。

【 0 1 7 6 】

この処理が終わると、制御は取得ライブラリ 3 7 から G a t e サーバ 2 7 に戻る。

【 0 1 7 7 】

G a t e サーバ 2 7 は、W e b サーバ 2 8 に対してユーザ 2 が指定した URL を通知し、A S P アプリケーション 3 3 によるサービスが提供される。サービスとしては、例えば発注システムの機能提供、スケジュール管理ツールの機能提供などがある。

【 0 1 7 8 】

続いて、図 1 2 に示すように、G a t e サーバ 2 7 は、情報設定メソッドを用いて、ベーシック連携サービスを提供する機能に渡され後処理で利用される情報を、取得ライブラリ 3 7 に引き渡す。引き渡される情報には、例えば課金を行うためのユーザの接続時間やログを記録するための時刻情報などがある。

【 0 1 7 9 】

次に、G a t e サーバ 2 7 は、取得ライブラリ 3 7 の後処理メソッドを呼び出す。すると、前処理の場合と同様に、取得ライブラリ 3 7 は、サービスアダプタ対応付け部 4 0 から必要なアダプタを取得し、該当するアダプタの処理メソッドを呼び出す。これにより、ベーシック連携サービス（この例では課金管理サー

ビス 3 1 b) が提供される。

【 0 1 8 0 】

図 1 3 は、アプリケーション側リクエスト取得機能 2 5 の前段動作の一例を示すシーケンス図である。この図 1 3 は、ユーザ 2 が電子メールサービス 3 5 a を使用するまでの処理について示している。このようにサービスアダプタ対応付け部 4 0 を用いることで、ASP 毎に、提供される付加サービスの種類を切り換えることができる。また、表 4 のようにテーブルで設定すれば、このテーブルを更新するだけで ASP に提供される付加サービスの種類を変更できるため IDC 業者 2 0 は管理・運用が容易になり、ASP に提供する付加サービスの種類に柔軟性を持たせることができる。

【 0 1 8 1 】

また、図 1 4 は、アプリケーション側リクエスト取得機能 2 5 の後段動作の一例を示すシーケンス図である。この図 1 4 は、ユーザ 2 が電子メールサービス 3 5 a の使用を終えた後の処理について示している。

【 0 1 8 2 】

この図 1 3 及び図 1 4 は、すでに ASP アプリケーション 3 3 が使われている時点から記述されている。また、図 1 3 及び図 1 4 は、ユーザ 2 からのメール送信のリクエストに応じるために電子メールサービス 3 5 a が使用される場合を例示している。

【 0 1 8 3 】

ASP アプリケーション 3 3 の処理の一部によってユーティリティ連携サービスの 1 つである電子メールサービス 3 5 a が使用される前に、ユーザ管理サービス 3 1 a の依頼側である ASP 1 8 の認証チェックが行われ、課金管理サービス 3 1 b により処理依頼側である ASP 1 8 の付加サービス使用料がチェックされる。

【 0 1 8 4 】

また、ASP アプリケーション 3 3 の処理の一部によってユーティリティ連携サービスの 1 つである電子メールサービス 3 5 a が使用された後に、課金管理サービス 3 1 b により依頼側である ASP 1 8 の課金情報が蓄積される。

【0185】

まず、図13に示すように、ASPアプリケーション33は電子メールサービスアダプタ34aを生成し、アダプタ34aに対してメール送信を要求する。

【0186】

すると、電子メールサービスアダプタ34aは、取得ライブラリ37に対して情報を渡す。

【0187】

取得ライブラリ37は、前処理メソッドを呼び出し、実際の電子メールサービス35aを提供する機能に対してメール送信処理を依頼し、図14に示すように後処理メソッドを呼び出す。

【0188】

取得ライブラリ37は、アダプタを取得するためのキーワードとしてサービス名「電子メール」に加え、使用するメソッド名「メール送信」も使用する。その他の取得ライブラリ37の動作は、ユーザ側リクエスト取得機能24の場合と同様である。ここでも、ユーザ側リクエスト取得機能24と同様に表4のようなサービスアダプタ対応付けテーブルを使用する。これにより、ASP18に提供される付加サービスの種類が固定されず、より柔軟性を持たせることができるため、iDC業者20はASP18の要望にあった付加サービスを提供できる。また、ASP18との契約内容に変更が生じてこのテーブルを更新するだけで継続使用可能であり、ライブラリ自体の再コンパイルが不要であるため、iDC業者20側の運用コストを軽減させることができる。

【0189】

なお、必要になった時点でそれぞれのアダプタを初期化する場合、アダプタの選択要求からサービス提供までに処理時間がかかる。したがって、事前にアダプタ群を生成してプーリング（保留）しておき、処理を高速化してもよい。

【0190】

以上のような情報処理サービス提供システムを実現すると、iDC業者20、ASP18は以下のような利点を得ることができる。

【0191】

iDC業者20は、iDC21上にアプリケーションプラットフォーム22を具備することにより、単にiDC21を使用するのみでは実現できない各種付加サービスを提供できる。これにより、iDC業者20は、異なる機能の使用を求めるそれぞれのASPに対して適切な機能を容易に提供できる。また、iDC業者20は、付加サービスを使用したユーザ2又はASP18からその使用に応じた対価を得ることができる。

【0192】

iDC業者20は、ASP18との契約内容に変更が生じた場合にその変更内容を容易に事業に反映させることができ、自己の提供するサービスを実現する機能の種類、サービスの内容を容易に追加／変更できる。これにより、iDC業者20は、事業の運用・保守コストを削減でき、利益効率を高めることができる。

【0193】

iDC業者20は、ASP18に対して他のASPの提供する付加サービスを提供することで、データセンタ単独での事業を行う場合よりも自データセンタの付加価値を高めることができ、自データセンタを使用するユーザ2、ASPを増加させることができる。

【0194】

ASP18は、自己の提供するASPサービスに付加サービスを連携させることができる。

【0195】

また、ASP18は、他のASPと連携してユーザ2にサービスを提供し、ユーザ2の使用可能な機能を増加させてユーザ2の利便性を増すことができ、ユーザ2に使用される機会を増加させることができる。

【0196】

また、ASP18は、付加サービスを提供するために必要な技術力を持たない場合であっても、他のASPによって提供される付加サービスを使用することで、容易にサービスの充実を図ることができる。

【0197】

また、ASP18は、本来ASPサービスの提供用として作られていないアプ

リケーションをASPサービス用として機能させる場合に、ASPとして機能するために必要な機能（課金管理やシステム監視などの機能）を付加サービスで補充できる。したがって、ソフトウェアのASPサービス提供用に容易に変更できる。

【0198】

また、ASP18は、他のASPとの間でユーザ管理や課金管理を共通化できるため、ユーザ2に対して複数のASPに対するシングルサインオン環境を提供でき、課金請求を統合できる。複数のASPに対するシングルサインオン環境や課金請求の統合は、ユーザ2にとっても管理が容易となるため、ユーザの獲得を促進できる。

【0199】

（第3の実施の形態）

上記各実施の形態に係る情報処理サービス提供システムにおける各機能、各要素は、同様の作用・動作を実現可能であれば配置を変更させてもよく、また各機能、各要素を自由に組み合わせてもよく、各機能、各要素を分割してもよい。

【0200】

図15は、本実施の形態に係る情報処理サービス提供システムの構成を例示するブロック図である。この図15は、上記図7に示す情報処理サービス提供システムの変形例を示す。

【0201】

ベーシック連携サービス31a～31cを提供する機能は、ベーシック連携アダプタ31a～31cを経由してリクエストを受け又は発する。

【0202】

ユーティリティ連携サービス35を提供する機能は、ユーティリティサービスアダプタ34を経由してリクエストを受け又は発する。

【0203】

リクエスト取得サーバ42は、リクエスト取得機能42a、42bとサービスアダプタ対応付け部40を具備する。

【0204】

リクエスト取得サーバ42は、Webブラウザ29、付加サービス31a～31c、35を提供する機能、ASPアプリケーション33のそれぞれからリクエストを取得し、サービスアダプタ対応付け部40の内容にしたがって取得したリクエストに応じて、ASPアプリケーション33又は付加サービス31a～31c、35を提供する機能を選択し、選択したASPアプリケーション33又は付加サービス31a～31c、35を提供する機能にリクエストを送る。

【0205】

これにより、あらゆる情報処理サービスが他の情報処理サービスを使用できる関係となり、サービス間の連携が効率的に実現される。その結果として、ASPアプリケーションと複数の付加的機能を組み合わせた機能の構築やカスタマイズが容易となる。

【0206】

例えば、付加的機能から他の付加的機能に処理実行を依頼でき、サービスを提供する各種機能を自由に連携させることができる。

【0207】

上記各実施の形態に係る情報処理サービス提供システムの各機能、各要素は、コンピュータに実行させることのできるプログラム、モジュール、プロセスとして提供できる。このプログラムにより上記各機能、各要素を提供する場合には、例えば磁気ディスク（フレキシブルディスク、ハードディスク等）、光ディスク（CD-ROM、DVD等）、半導体メモリなどの記録媒体16にそのプログラムを書き込んでコンピュータに適用可能である。また、このプログラムは通信媒体によりコンピュータ、コンピュータシステムに伝送しても適用可能である。コンピュータは、記録媒体16に記録されたプログラムを読み込み、そのプログラムによって動作が制御されることにより、上述した処理を実行する。

【0208】

【発明の効果】

以上詳記したように本発明においては、情報処理サービスに対して付加的な情報処理サービスを連携させることができる。これにより、情報処理サービスの提供を容易化できる。また、情報処理サービスの内容を充実させることができる。

【図面の簡単な説明】

【図 1】

本発明の第 1 の実施の形態に係る情報処理サービス提供システムの構成の一例を示すブロック図。

【図 2】

同実施の形態に係る情報処理サービス提供システムの構成の変形例を示すブロック図。

【図 3】

本発明の第 2 の実施の形態に係る情報処理サービス提供システムの連携状態の一例を示すブロック図。

【図 4】

同実施の形態に係る情報処理サービス提供システムの構成の一例を示すブロック図。

【図 5】

ユーザ側リクエスト取得機能による連携状態の一例を示すブロック図。

【図 6】

アプリケーション側リクエスト取得機能による連携状態の一例を示すブロック図。

【図 7】

リクエスト取得機能を含む情報処理サービス提供システムのアーキテクチャの一例を示すブロック図。

【図 8】

同実施の形態に係る情報処理サービス提供システムのアーキテクチャと具体的なサービスとの関係の一例を示すブロック図。

【図 9】

リクエスト取得機能の構造の一例を示すクラス図。

【図 1 0】

リクエスト取得機能の構造と具体的なサービスとの関係の一例を示すクラス図。

【図 1 1】

ユーザ側リクエスト取得機能の前段動作の一例を示すシーケンス図。

【図 1 2】

ユーザ側リクエスト取得機能の後段動作の一例を示すシーケンス図。

【図 1 3】

アプリケーション側リクエスト取得機能の前段動作の一例を示すシーケンス図

【図 1 4】

アプリケーション側リクエスト取得機能の後段動作の一例を示すシーケンス図

【図 1 5】

本発明の第 3 の実施の形態に係る情報処理サービス提供システムの構成の一例を示すブロック図。

【符号の説明】

3 … クライアント

5 a、5 b … A S P サーバ

6 a、6 b … アプリケーション

7 … イベント検出機能

8 … ライブラリ

9 … 参照機能

1 0、1 5、2 4 b、2 5 b … 依頼機能

1 2 a、1 2 b … 付加サーバ

1 3 … 記録機能

1 4 … データベース

1 7 … A S P ポータル

1 8 … A S P

1 9 … 付加サービス業者

2 0 … i D C 業者

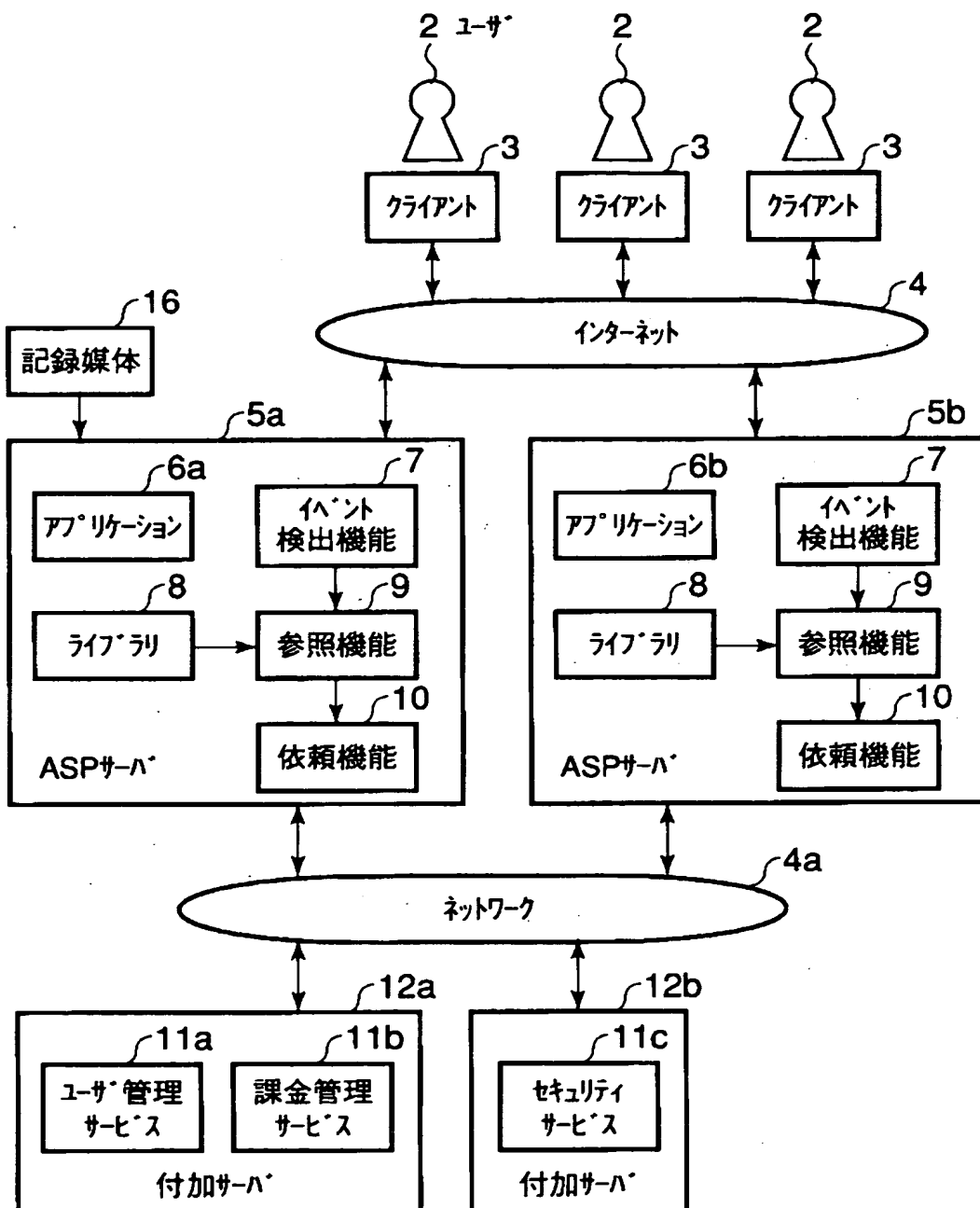
2 1 … i D C

- 2 2 …アプリケーションプラットフォーム
- 2 4 …ユーザ側リクエスト取得機能
- 2 4 a、2 5 a、4 2 a、4 2 b …リクエスト取得機能
- 2 5 …アプリケーション側リクエスト取得機能
- 3 0、3 0 a～3 0 c …ベーシックサービスアダプタ
- 3 1、3 1 a～3 1 c …ベーシック連携サービス
- 3 4 …ユーティリティサービスアダプタ
- 3 5 …ユーティリティ連携サービス
- 3 7 …取得ライブラリ
- 3 8 …ベーシックサービスアダプタ規約
- 3 9 …ユーティリティサービスアダプタ規約
- 4 0 …サービスーアダプタ対応付け部
- 4 1 …サービスーURL対応付け部
- 4 2 …リクエスト取得サーバ

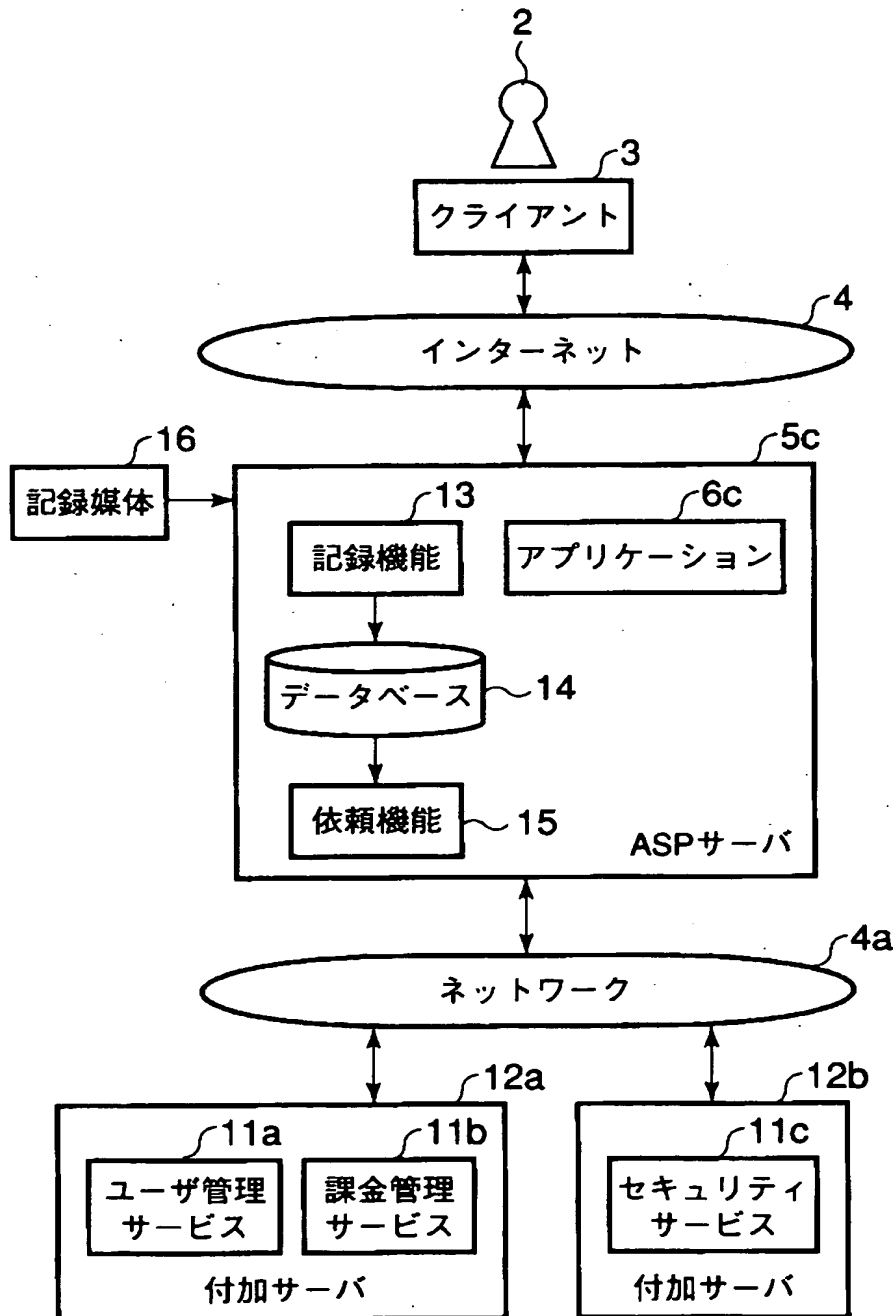
【書類名】

図面

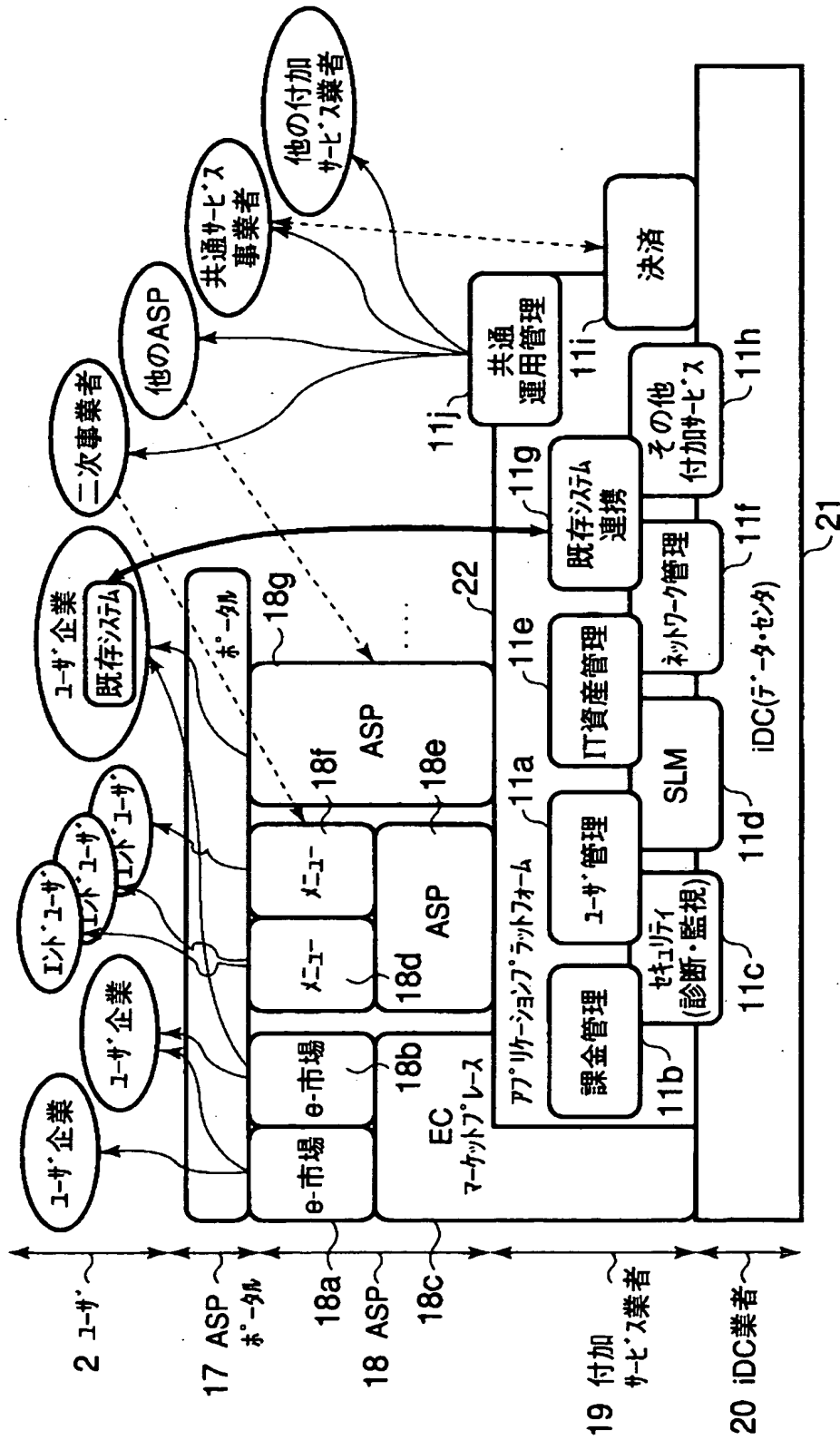
【図 1】



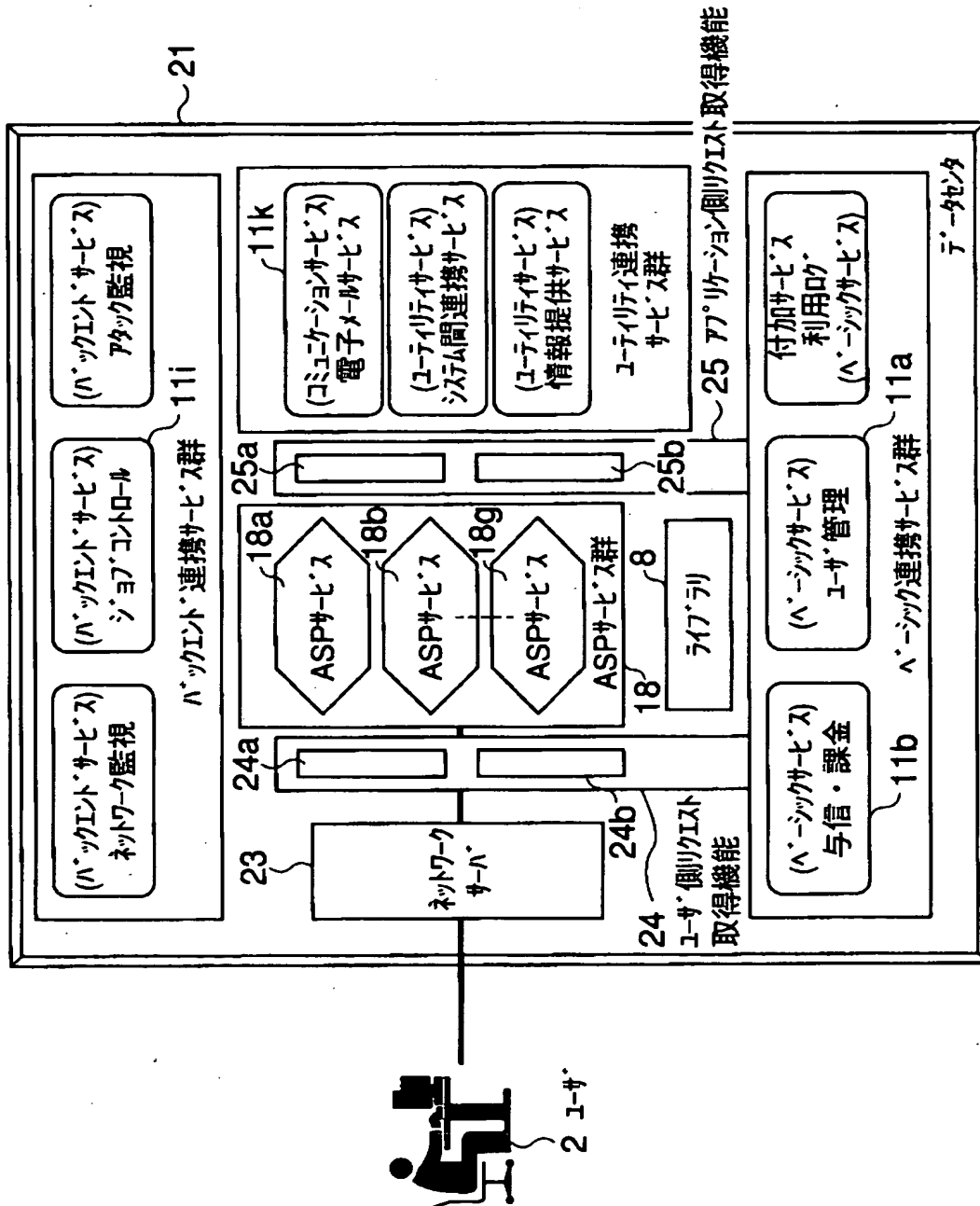
【図 2】



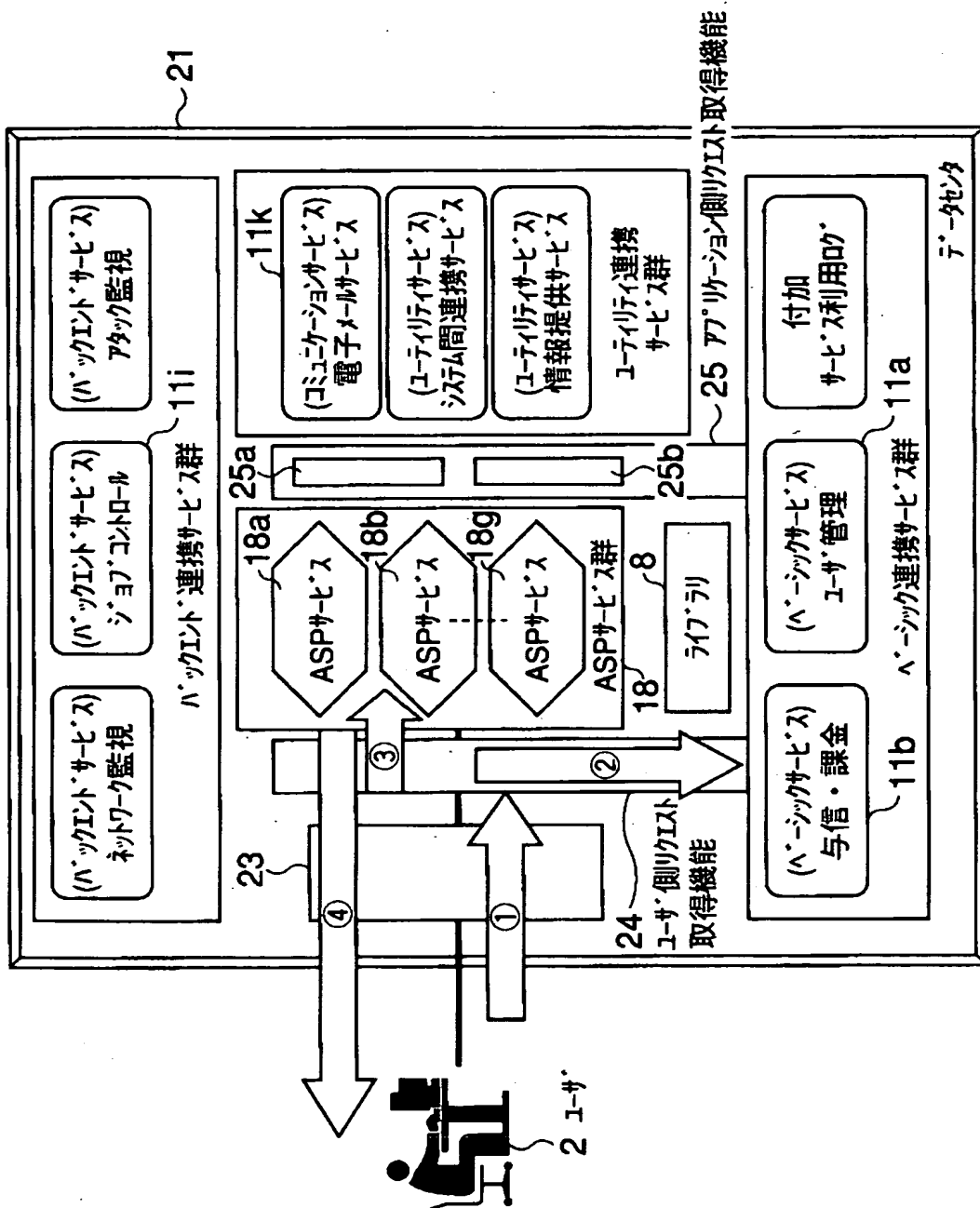
【図 3】



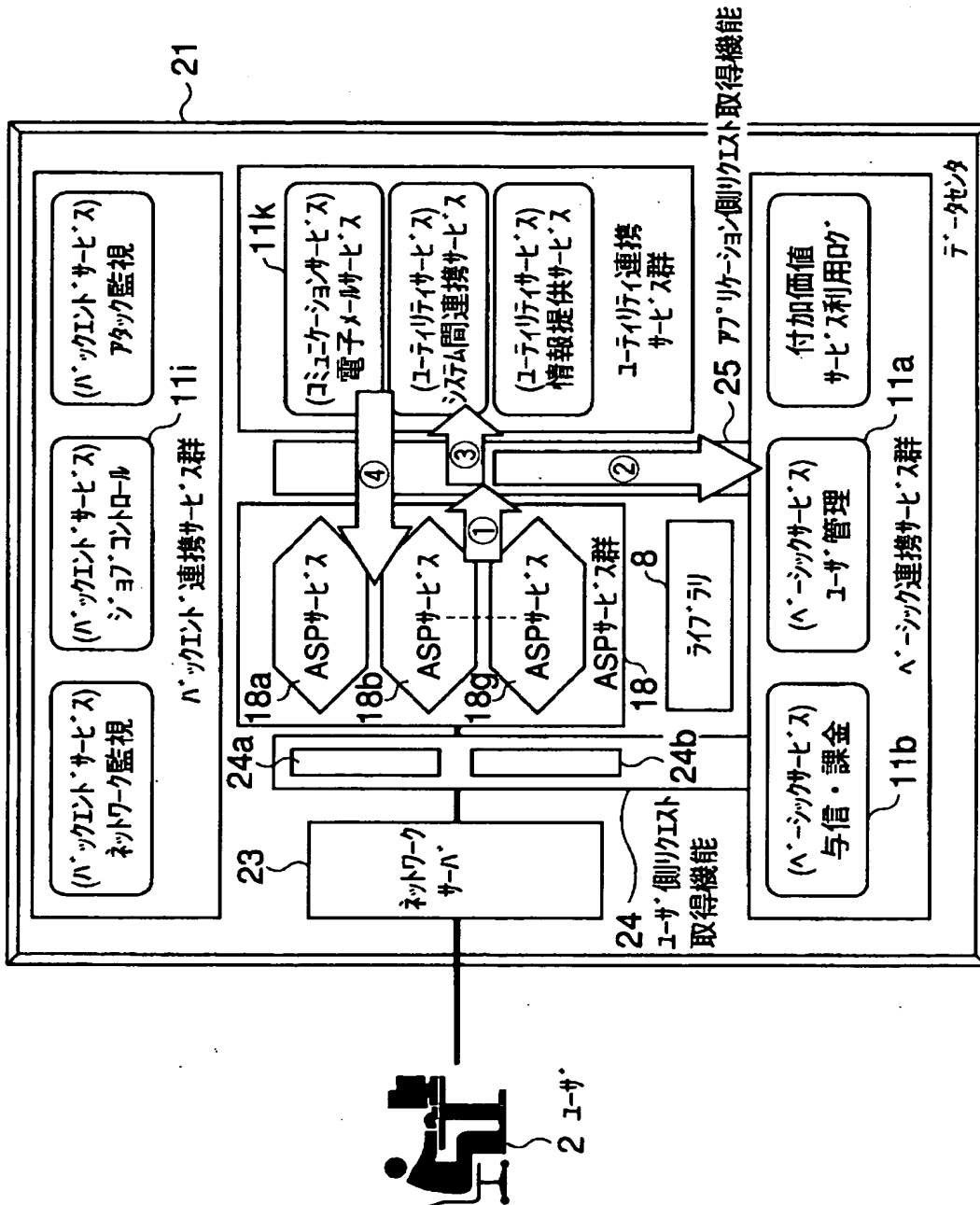
【図 4】



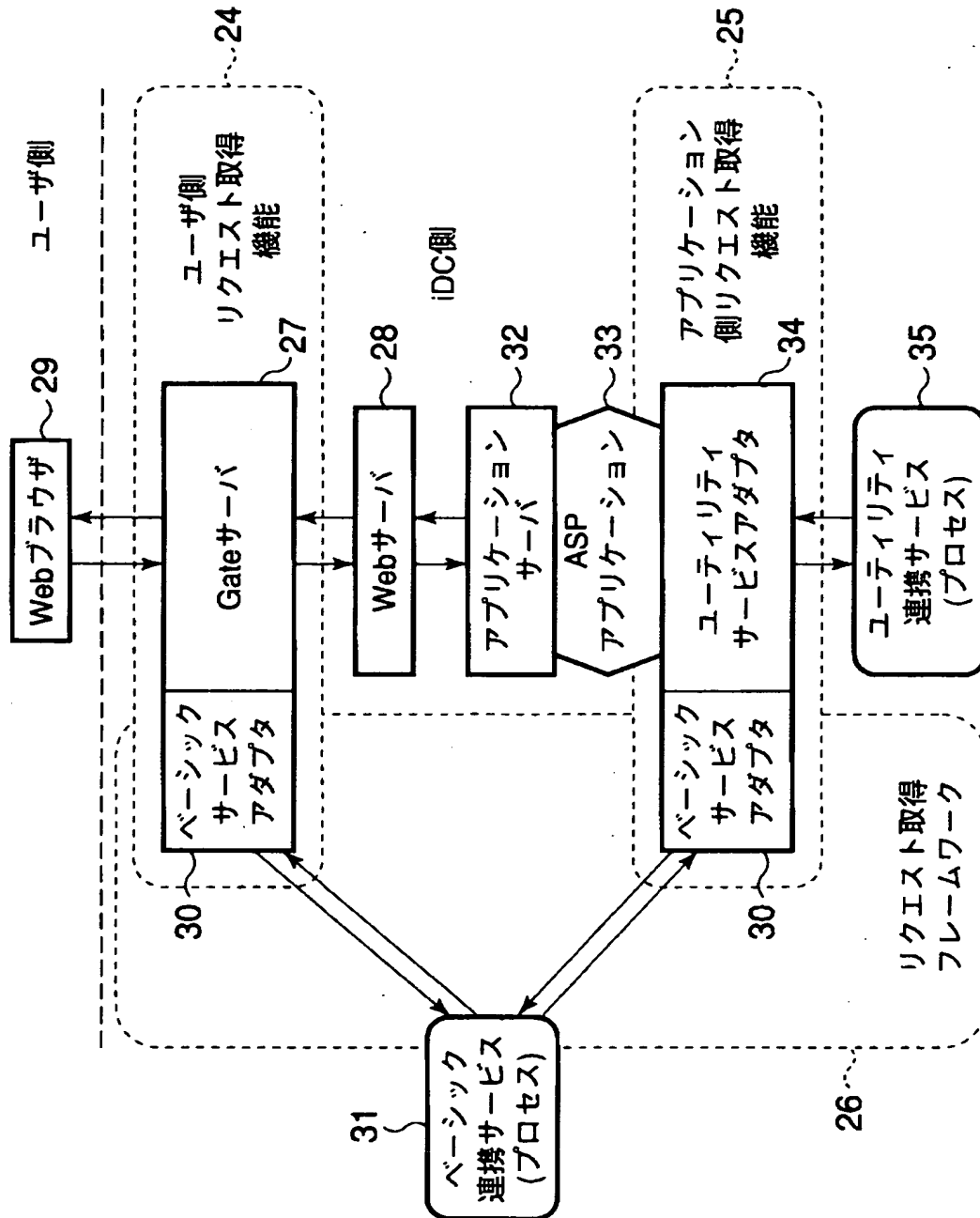
【図5】



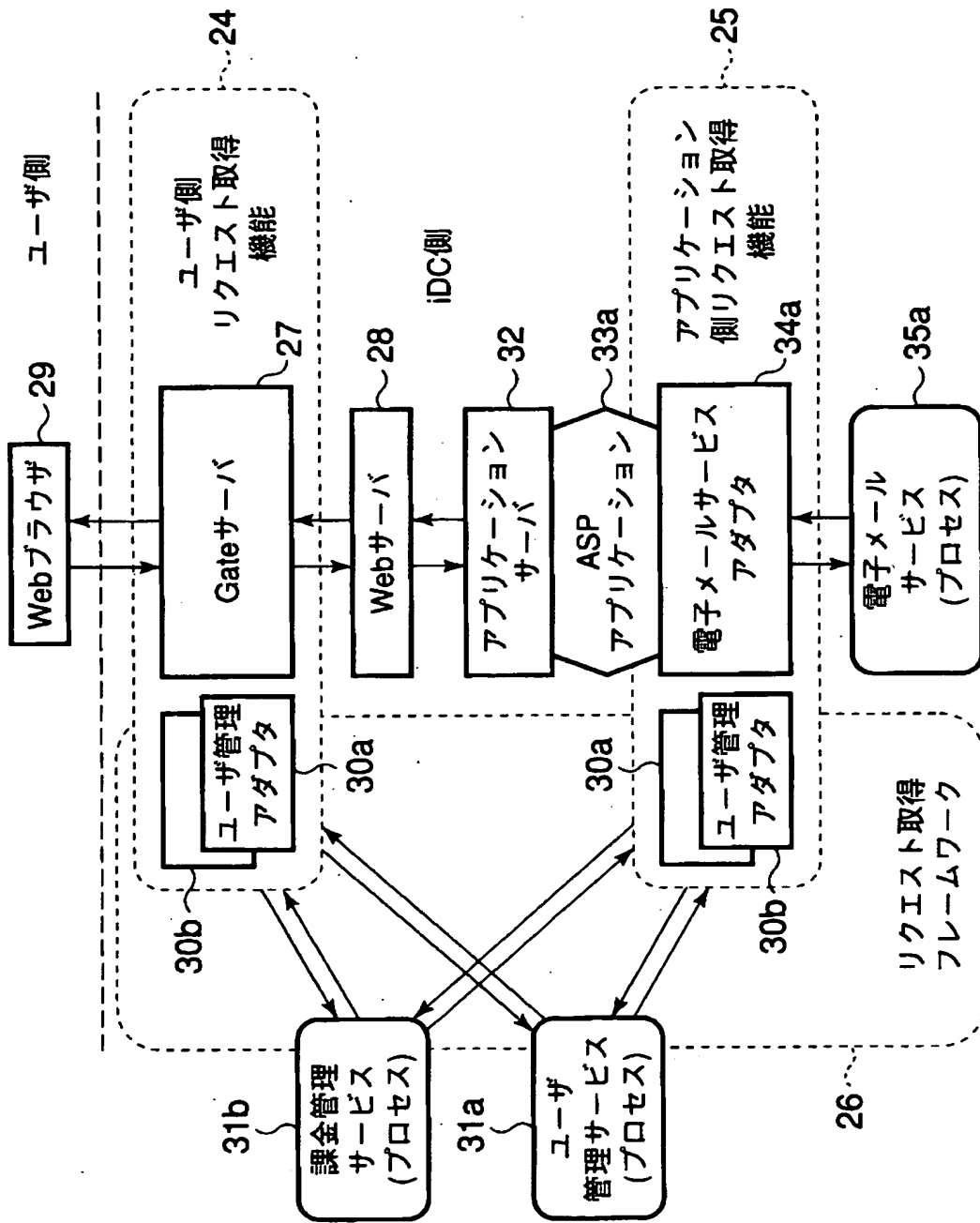
【図6】



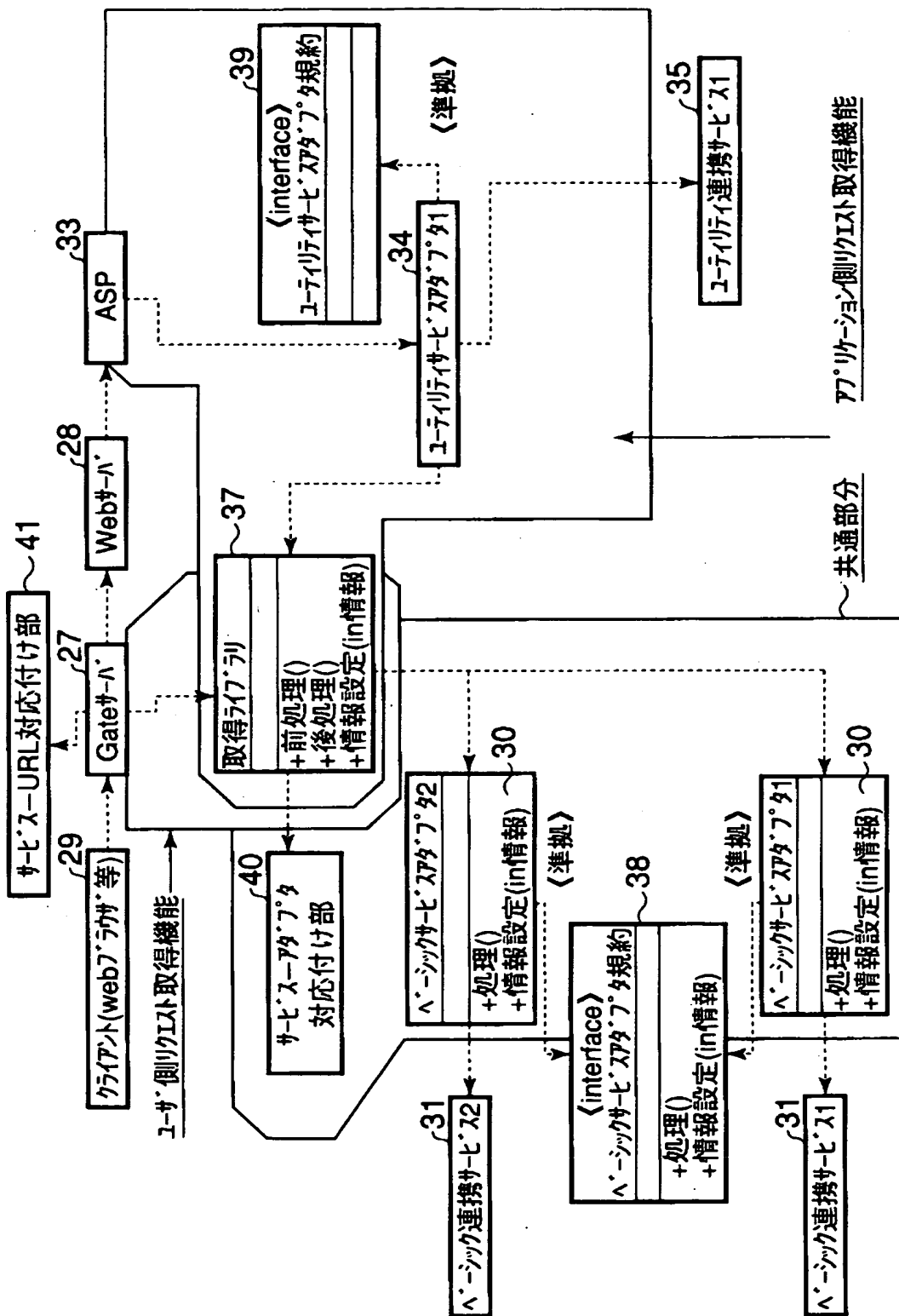
【図7】



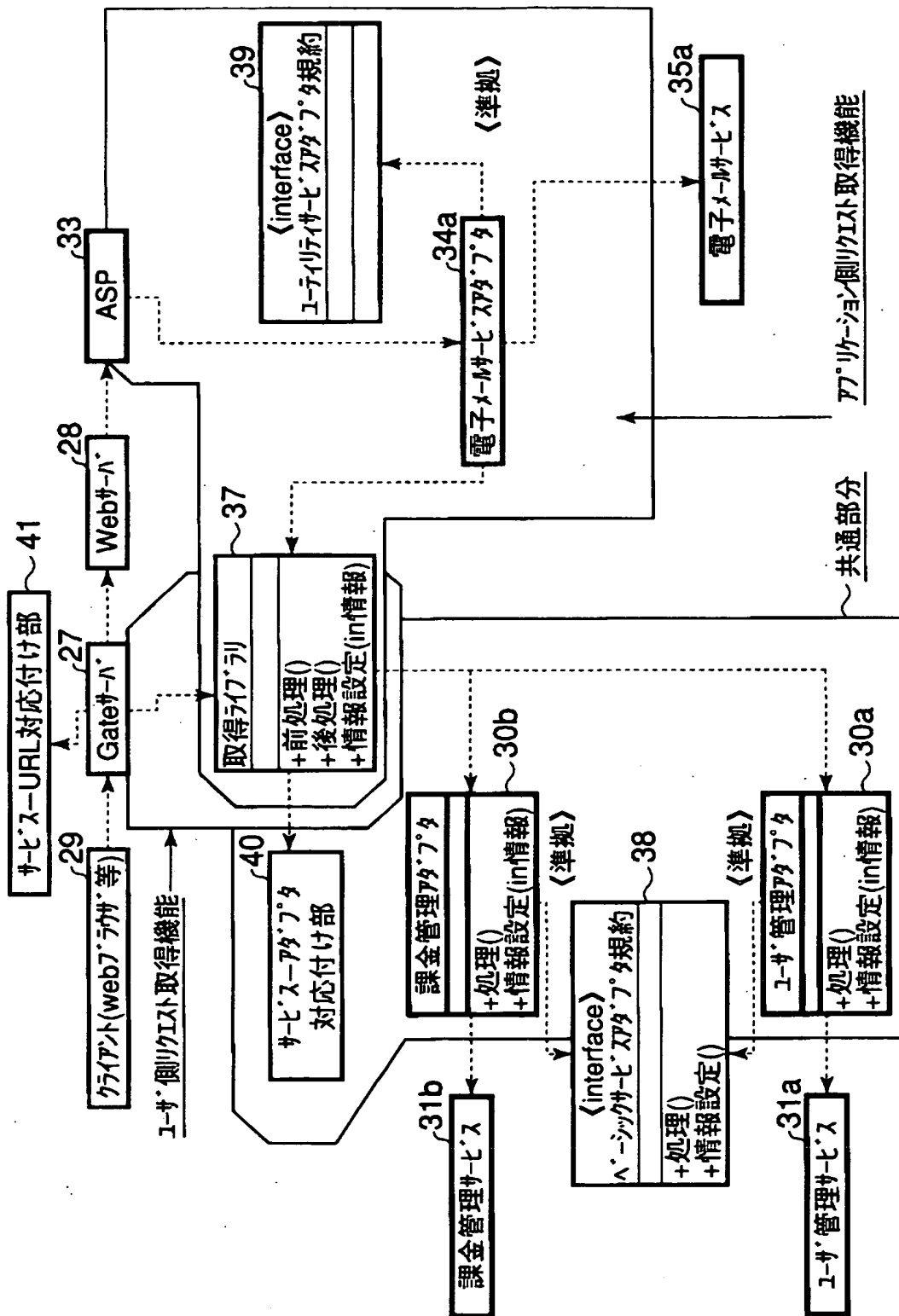
【図 8】



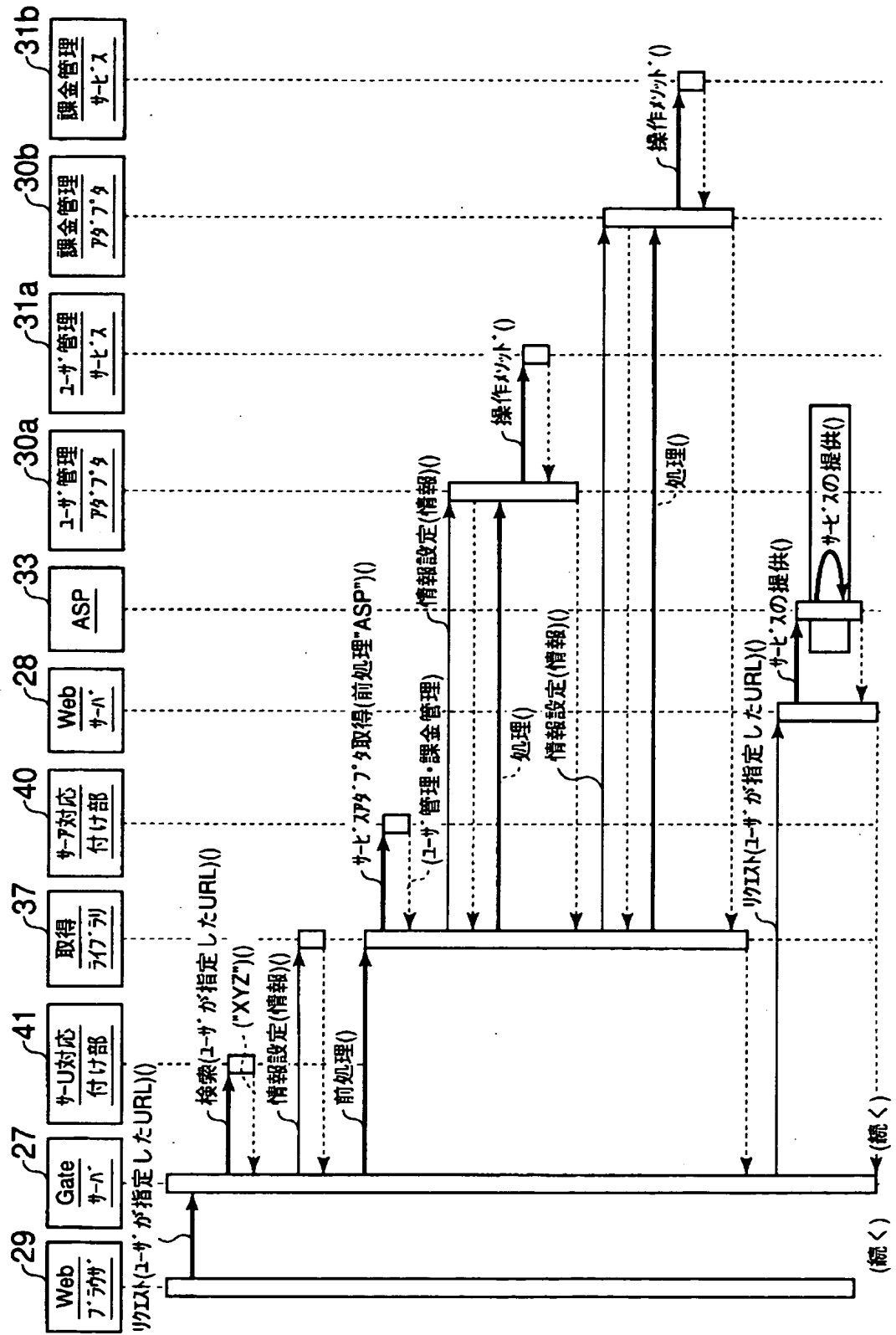
【图9】



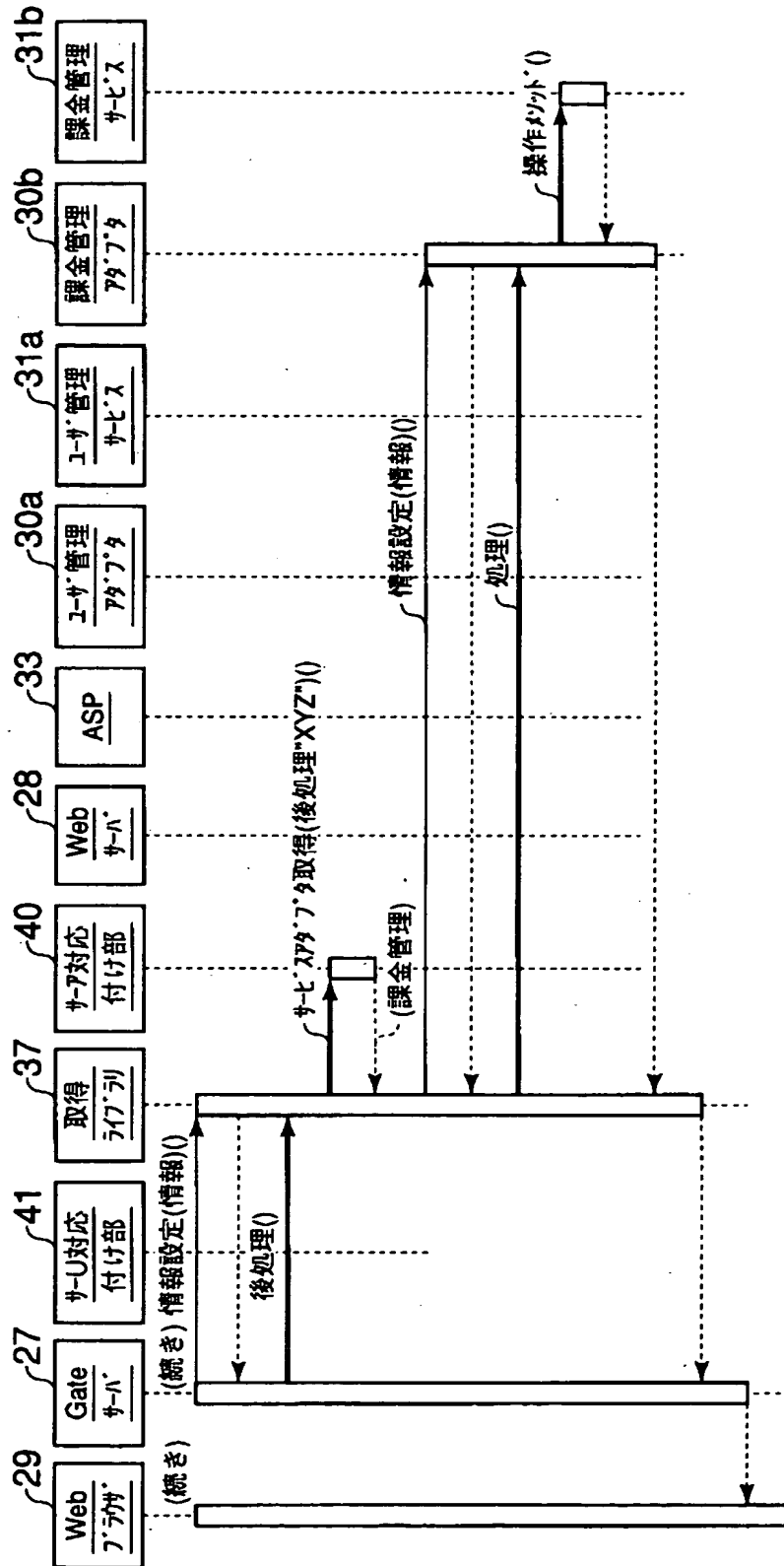
【図10】



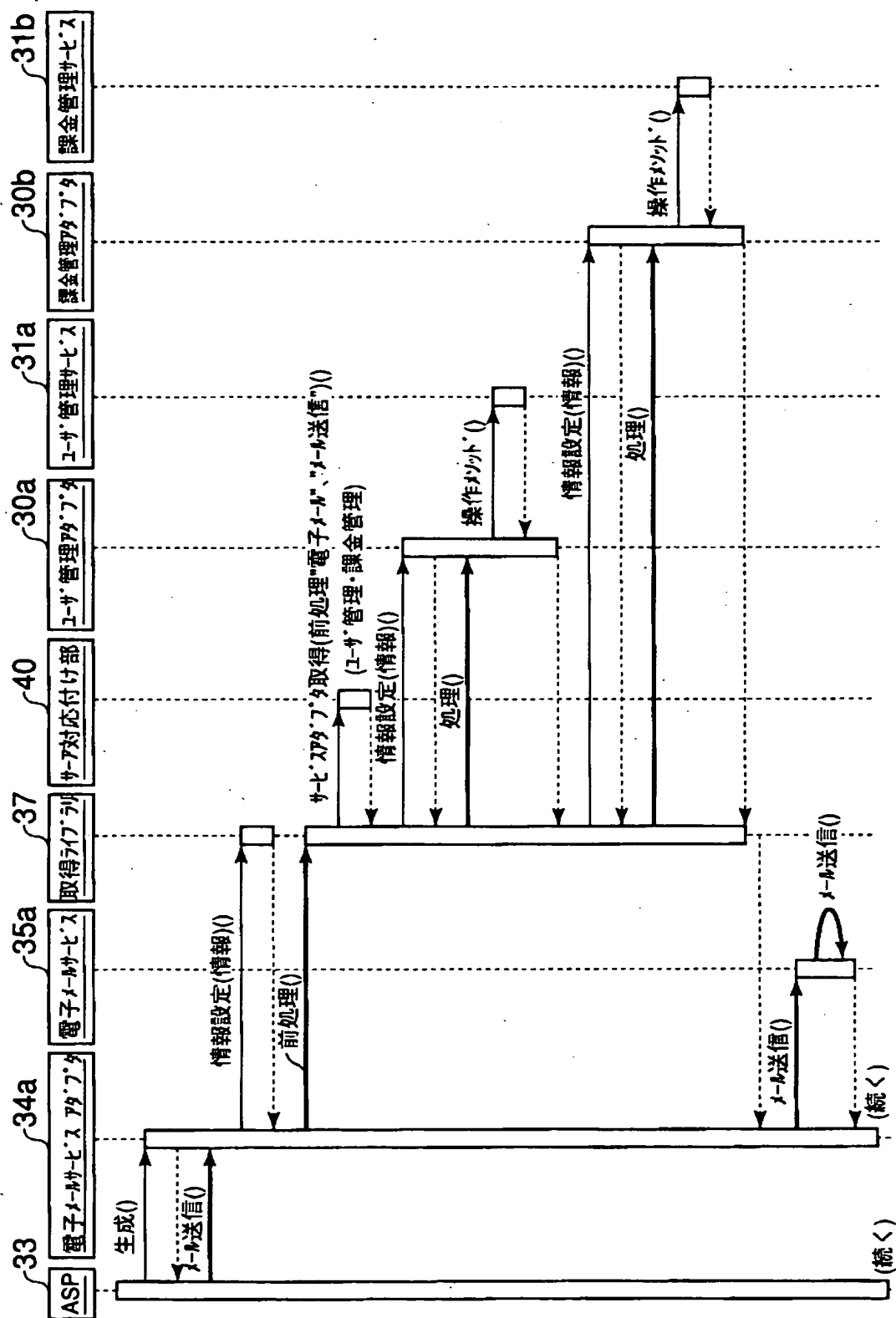
【図 1 1】



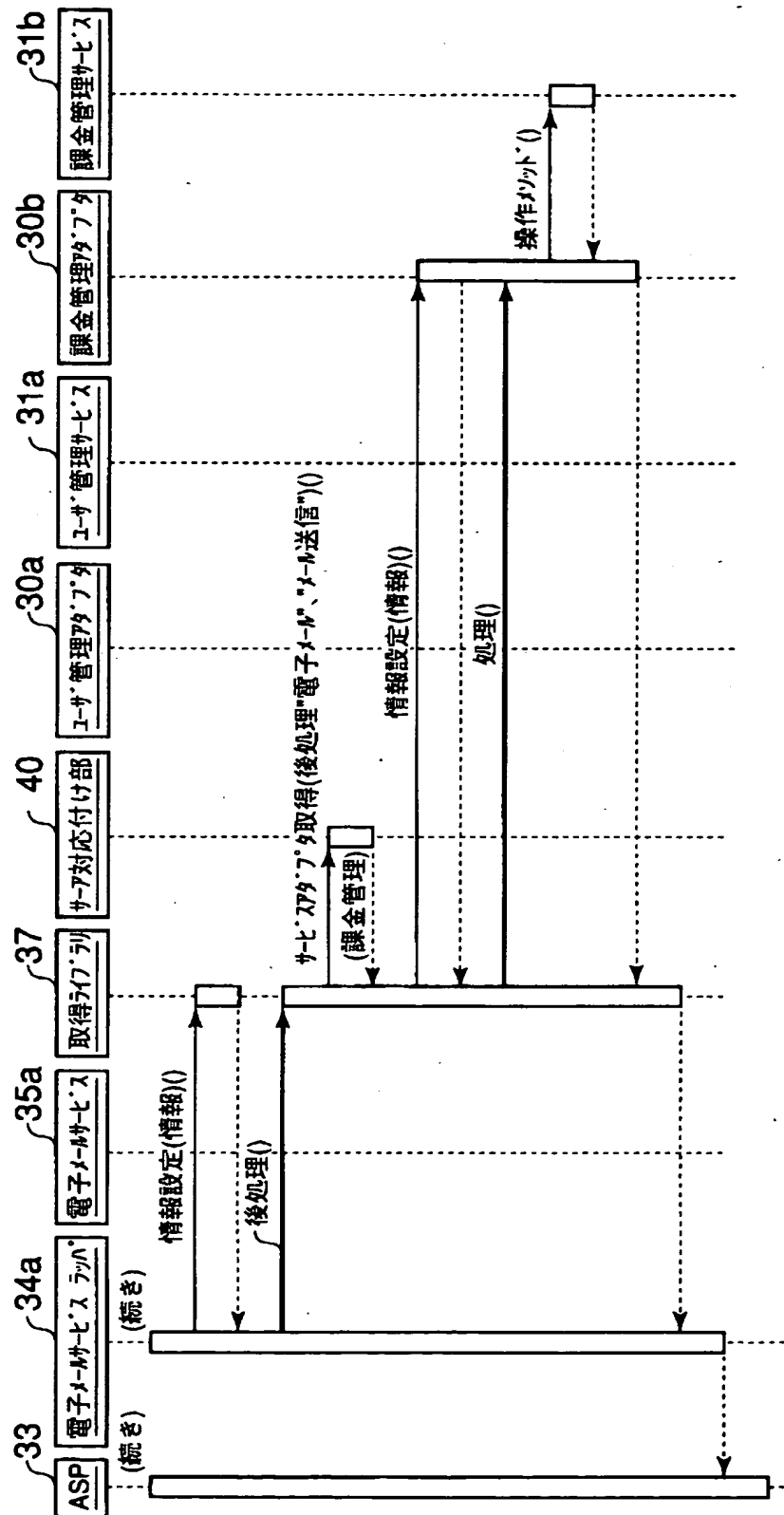
【図 1 2】



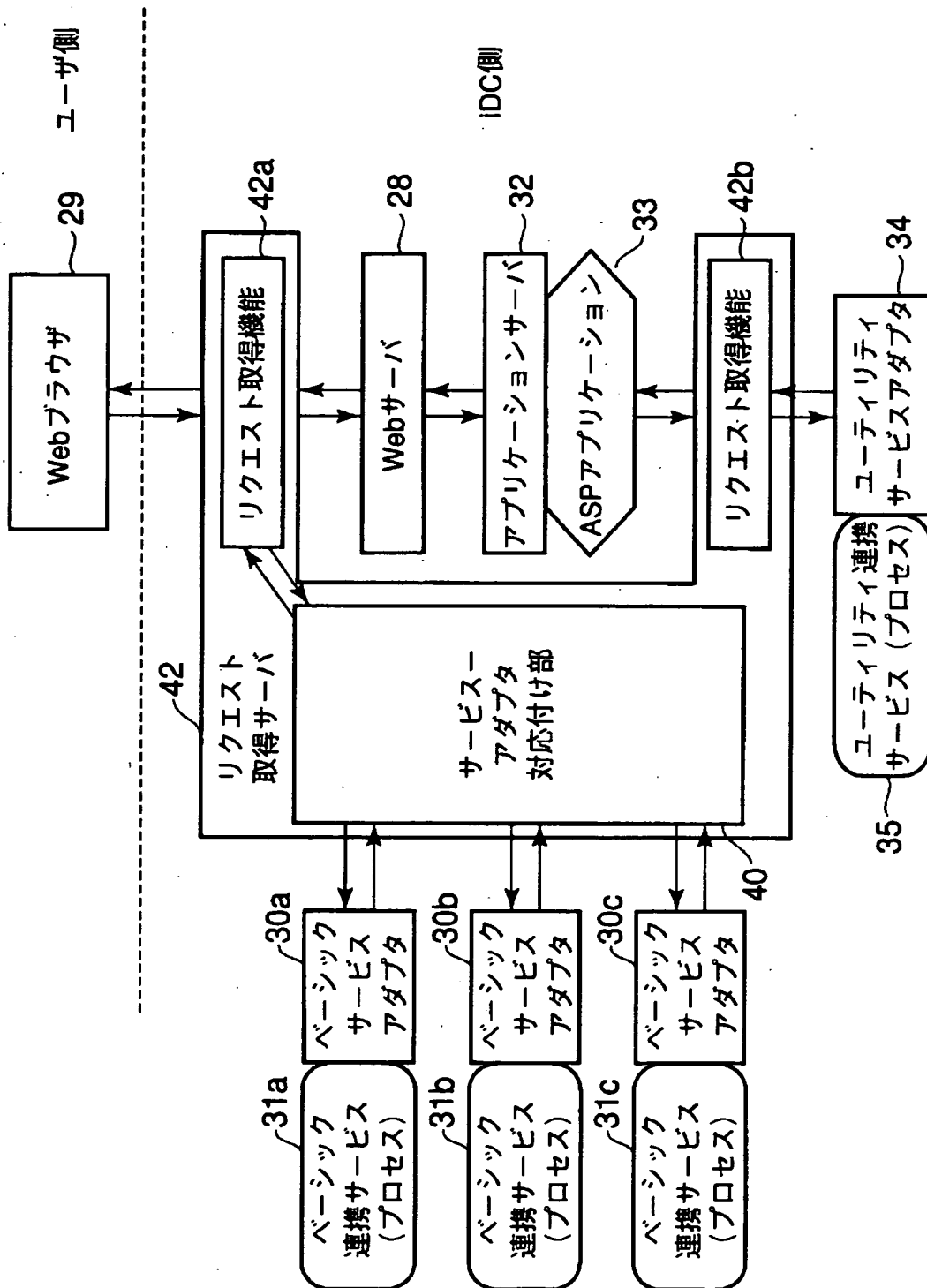
【図 13】



【図14】



【図15】



【書類名】 要約書

【要約】

【課題】 A S P サービスを容易に実施可能とする。

【解決手段】 コンピュータによる情報処理サービス提供方法に関する発明である。本発明の情報処理サービス提供方法では、ネットワークを経由して提供される情報処理サービスに対する付加的な情報処理サービスを依頼するイベントを検出し、イベントが検出された場合に、付加的な情報処理サービスを提供する付加的機能の呼び出しを定義したライブラリを参照し、ライブラリの参照結果に基づいて、付加的な情報処理サービスを提供する付加的機能に実行を依頼する。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願2002-069188
受付番号	50200353563
書類名	特許願
担当官	第七担当上席 0096
作成日	平成14年 3月18日

<認定情報・付加情報>

【特許出願人】

【識別番号】	000003078
【住所又は居所】	東京都港区芝浦一丁目1番1号
【氏名又は名称】	株式会社東芝

【代理人】

申請人

【識別番号】	100058479
【住所又は居所】	東京都千代田区霞が関3丁目7番2号 鈴榮特許 綜合法律事務所内
【氏名又は名称】	鈴江 武彦

【選任した代理人】

【識別番号】	100084618
【住所又は居所】	東京都千代田区霞が関3丁目7番2号 鈴榮特許 綜合法律事務所内
【氏名又は名称】	村松 貞男

【選任した代理人】

【識別番号】	100068814
【住所又は居所】	東京都千代田区霞が関3丁目7番2号 鈴榮特許 綜合法律事務所内
【氏名又は名称】	坪井 淳

【選任した代理人】

【識別番号】	100092196
【住所又は居所】	東京都千代田区霞が関3丁目7番2号 鈴榮國特 許綜合法律事務所内
【氏名又は名称】	橋本 良郎

【選任した代理人】

【識別番号】	100091351
【住所又は居所】	東京都千代田区霞が関3丁目7番2号 鈴榮特許 綜合法律事務所内

次頁有

認定・付加情報（続き）

【氏名又は名称】	河野 哲
【選任した代理人】	
【識別番号】	100088683
【住所又は居所】	東京都千代田区霞が関3丁目7番2号 鈴榮特許 綜合法律事務所内
【氏名又は名称】	中村 誠
【選任した代理人】	
【識別番号】	100070437
【住所又は居所】	東京都千代田区霞が関3丁目7番2号 鈴榮特許 綜合法律事務所内
【氏名又は名称】	河井 将次